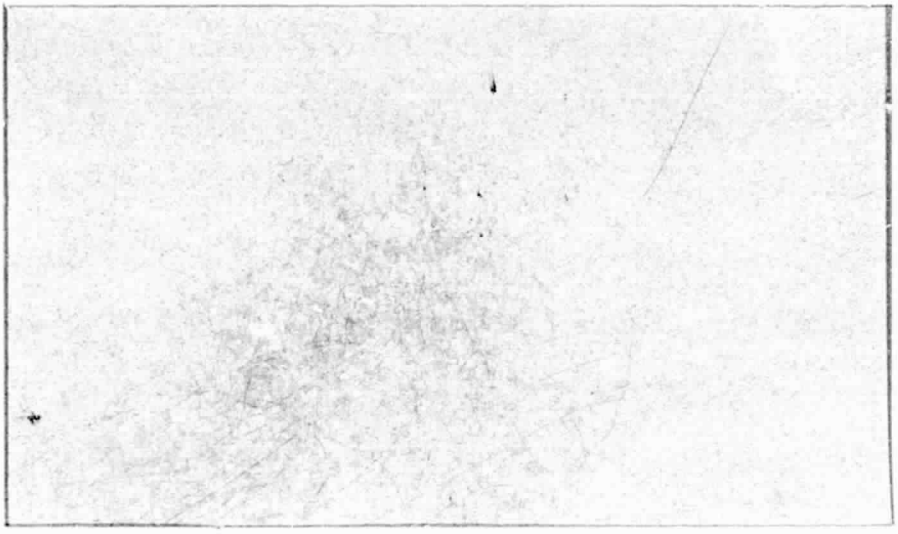


(NASA-CR-162911) ALTERNATIVES FOR JET ENGINE CONTROL Semiannual Status Report, 1 Sep. 1979 - 29 Feb. 1980 (Notre Dame Univ.) 93 p HC A05/MF A01 CSCL 21E

N80-20273

Unclas
G3/07 46739

*Department of*

# ELECTRICAL ENGINEERING

## UNIVERSITY OF NOTRE DAME, NOTRE DAME, INDIANA

Semi-Annual Status Report

to the

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

on

NASA Grant NSG-3048

# ALTERNATIVES FOR JET ENGINE CONTROL*

September 1, 1979 - February 29, 1980

*This work has been carried out under the direction of

Dr. Michael K. Sain
Department of Electrical Engineering
University of Notre Dame
Notre Dame, Indiana
USA 46556

TABLE OF CONTENTS

# 1. SUMMARY

This report is a statement of progress on NASA Grant NSG-3048 during the six month period from September 1, 1979 to February 29, 1980. During this period, the researches at the University of Notre Dame were directed by Professor Michael K. Sain; and the funded research assistant was Mr. Stephen Yurkovich. Mr. R. Michael Schafer continued studies developing out of researches under this grant; his support, however, was drawn from fellowship funds made available by the University of Notre Dame. Overseeing technical aspects of the grant at Lewis Research Center was Dr. Kurt Seldner.

The major emphasis of this status report lies in the continuation of nonlinear modeling researches involving the use of tensor analysis. Progress has been achieved by extending the studies to the controlled equation

$$\dot{x} = f(x,u)$$

and by considering more complex situations. Included herein are calculations illustrating the modeling methodology for cases in which $x$ and $u$ take values in real spaces of dimension up to three, and in which the degree of tensor term retention is as high as three. The quality of the controlled tensor models has been most encouraging; and preparations are now under way to begin applications to the QCSEE digital simulation.

Though not funded by Grant NSG-3048 during this period, certain linear multivariable studies growing out of earlier grant work are described briefly, for completeness.

## 2. STATUS OF CURRENT RESEARCH

This section reviews the status of grant researches carried out during the six-month period from September 1, 1979 to February 29, 1980.

In the previous Semi-Annual Status Report, for the period from March 1, 1979 to August 31, 1979, the initial studies on nonlinear modeling by means of tensor ideas were presented. Basically, those results were concerned with homogeneous systems of the form

$$\dot{x} = f(x),$$

for $x$ a member of some finite dimensional real vector space $X$. If

$$f(0) \neq 0,$$

then the equilibrium point $x_e$ satisfying

$$f(x_e) = 0$$

can be translated to the origin. Without loss then, it is assumed that

$$f(0) = 0.$$

Then $f(x)$ is to be understood in terms of its power series expansion about the origin, with each real component

$$f_i : X \to R$$

of the function

$$f : X \to X$$

leading to series terms of the form

$$x_{k_1} x_{k_2} \cdots x_{k_p} .$$

Such terms, though nonlinear, are nonetheless p-linear functions

$$R^p \to R,$$

which by basic tensor theory can be expressed in terms of linear functions

on the tensor product (p times) of R with itself. In the aggregate, the collection of such linear functions can be understood in terms of a matrix operator on the tensor product (p times) of X with itself. This matrix operator can be derived, if f is known, or determined empirically if f is not. The preceding Semi-Annual Status Report considered both viewpoints.

For control problems, of course, interest would center on an equation of the form

$$\dot{x} = f(x,u),$$

where u is a member of another finite dimensional real vector space U. The pair

$$(x,u)$$

is then an element of the product space

$$X \times U,$$

which is sometimes denoted

$$X \oplus U.$$

The focus of this report is on this case, and the details are presented in Section 2.1 following.

## 2.1 Higher Order Linearizations of Nonhomogeneous Systems (S. Yurkovich)

All discussion and examples to this point have concerned systems with homogeneous nonlinear vector differential equation representations. While emphasis has been centered around the linearization of nonlinear systems based on identification of the total operator L, the real thrust of the concept has been on identification of the individual linear operators associated with each tensor product term. It is this vein that the discussion will now pursue, with the inclusion of control (or input) variables in the system. Certainly the individual linear operators can be obtained from the partitions of the large matrix operator if so desired. Let it suffice to say that either could be considered a "by-product" of the other.

The initial venture of this chapter is to establish the notation and basic structure of the necessary tensor algebra for inclusion of forcing functions in the system. With this framework secured, the existing technique is adapted to accommodate nonhomogeneous multivariable systems; here a concept of basis ordering is introduced. To exhibit the application of the ideas developed, several examples are treated and resulting simulations plotted. First, two second-order systems are examined--that is, with two states and two controls. Progressing from these, a three state, three control example is inspected. Throughout, the choices for the forcing functions are sine wave inputs, with amplitude and frequency varied for each input. Again, intuition fails in these nonlinear equations as to the

choice of initial conditions; therefore, small initial conditions are chosen to ensure compliance with the feasible operating region of each system.

To distinguish between the concepts in this chapter and those developed previously, the usual symbols for the state vector and control vector will be employed: $x$ and $u$, respectively.

### 2.1.1 Formulation: Forcing Functions Included

The aim of this section is to extend the concepts treated thus far (the multilinearity of the higher degree nonlinear terms, for example) to functions of two variables, namely states and controls. With the notation convention established here, the software adaptation and representative examples can be discussed in the sequel.

Consider the general set of nonlinear vector differential equations

$$\dot{x} = f(x,u)$$

where the state vector $x$ is an element of $R^n$ and the input vector $u$ is an element of $R^m$. Define the Jacobian matrices

$$[\frac{\partial f}{\partial x}]_{ij} = \frac{\partial f_i}{\partial x_j} ,$$

$$[\frac{\partial f}{\partial u}]_{ij} = \frac{\partial f_i}{\partial u_j} ;$$

the Hessian matrices

$$\frac{\partial^2 f_i}{\partial x^2} , \quad i = 1,2,\ldots,n$$

as

$$[\frac{\partial^2 f_i}{\partial x^2}]_{jk} = \frac{\partial^2 f_i}{\partial x_j \partial x_k} ;$$

and the cross derivative matrices

$$\frac{\partial^2 f_i}{\partial x \partial u} \quad , \quad i = 1,2,\ldots,n$$

by

$$\left[\frac{\partial^2 f_i}{\partial x \partial u}\right]_{jk} = \frac{\partial^2 f_i}{\partial x_j \partial u_k} .$$

With a fixed operating point $(x_0,u_0)$, the general equation may be written as

$$\begin{aligned} f(x,u) = {} & f(x_0,u_0) \\ & + \left.\frac{\partial f}{\partial x}\right|_{(x_0,u_0)} (x-x_0) \\ & + \left.\frac{\partial f}{\partial u}\right|_{(x_0,u_0)} (u-u_0) \\ & + 1/2 \sum_{i=1}^{n} e_i (x-x_0)' \left.\frac{\partial^2 f_i}{\partial x^2}\right|_{(x_0,u_0)} (x-x_0) \\ & + 1/2 \sum_{i=1}^{n} e_i (u-u_0)' \left.\frac{\partial^2 f_i}{\partial u^2}\right|_{(x_0,u_0)} (u-u_0) \\ & + \sum_{i=1}^{n} e_i (x-x_0)' \left.\frac{\partial^2 f_i}{\partial x \partial u}\right|_{(x_0,u_0)} (u-u_0) \\ & + \ldots \end{aligned}$$

where (') denotes transposition and $e_i$ is the real vector of zeros except for the i-th entry which is unity.

Suppose now that

$$\frac{dx_0}{dt} = f(x_0,u_0),$$

so that by the substitutions

$$z(t) = x(t) - x_0(t) ,$$

$$w(t) = u(t) - u_0(t) ,$$

it follows that

$$\frac{dz}{dt} = \dot{x} - f(x_0,u_0) .$$

Therefore, the original differential equation is now converted to the following simplified form:

$$\dot{z} = \left.\frac{\partial f}{\partial x}\right|_{(x_0,u_0)} z + \left.\frac{\partial f}{\partial u}\right|_{(x_0,u_0)} w$$
$$+ 1/2 \sum_{i=1}^{n} e_i z' \left.\frac{\partial^2 f_i}{\partial x^2}\right|_{(x_0,u_0)} z$$
$$+ 1/2 \sum_{i=1}^{n} e_i w' \left.\frac{\partial^2 f_i}{\partial u^2}\right|_{(x_0,u_0)} w$$
$$+ \sum_{i=1}^{n} e_i z' \left.\frac{\partial^2 f_i}{\partial x \partial u}\right|_{(x_0,u_0)} w$$
$$+ \ldots$$

To introduce the notions of the tensor algebra, consider for instance the term

$$\sum_{i=1}^{n} e_i z' \left.\frac{\partial^2 f_i}{\partial x^2}\right|_{(x_0,u_0)} z$$

and regard it as a bilinear function

$$\sum_{i=1}^{n} e_i ( )' \left.\frac{\partial^2 f_i}{\partial x^2}\right|_{(x_0,u_0)} ( ).$$

Because of the bilinearity, this function can be expressed in terms of

the tensor product as a linear function operating on $z \otimes z$. Call this particular linear operator $L_{20}$ so that

$$1/2 \sum_{i=1}^{n} e_i z' \left.\frac{\partial^2 f_i}{\partial x^2}\right|_{(x_0,u_0)} z = 1/2\, L_{20}\,(z \otimes z).$$

The notation for the linear operator is such that the first subscript corresponds to the first vector, $z$, and the second subscript corresponds to $w$. Thus, by this convention,

$$\sum_{i=1}^{n} e_i z' \left.\frac{\partial^2 f_i}{\partial x \partial u}\right|_{(x_0,u_0)} w = L_{11}\,(z \otimes w).$$

Now define

$$L_{10} = \left.\frac{\partial f}{\partial x}\right|_{(x_0,u_0)},$$

$$L_{01} = \left.\frac{\partial f}{\partial u}\right|_{(x_0,u_0)},$$

as the usual "A" and "B" matrices of a standard linearization

$$\dot{z} = Az + Bw.$$

With this established, the differential equation can now be expressed as

$$\dot{z} = L_{10}\, z + L_{01}\, w + 1/2\, L_{20}\,(z \otimes z) + 1/2\, L_{02}\,(w \otimes w) + L_{11}\,(z \otimes w) + \ldots$$

The general term in such an expansion can be represented as

$$\frac{1}{j!k!} L_{jk} \left(\underbrace{z \otimes z \otimes \ldots \otimes z}_{j \text{ times}} \otimes \underbrace{w \otimes w \otimes \ldots \otimes w}_{k \text{ times}}\right),$$

so that, finally,

$$\dot{z} = \sum_{j=1}^{\infty} \sum_{k=0}^{\infty} \frac{1}{j!k!} L_{jk} (\underbrace{z \otimes \ldots \otimes z}_{j \text{ times}} \otimes \underbrace{w \otimes \ldots \otimes w}_{k \text{ times}}) .$$

Again, the motivation for such a derivation is the fact that all of the $L_{jk}$ are <u>linear</u> operators. The number of these operators for a given expansion depends, of course, on the desired accuracy of the approximation. Consider, for example, the table presented in Figure 2.1 for the typical system consisting of the state vector $x$ and the control vector $u$.

| <u>linear operators</u> | <u>terms retained in approximation</u> |
|---|---|
| $L_{10}$ $L_{01}$ | standard linearization |
| $L_{20}$ $L_{11}$ $L_{02}$ | second degree |
| $L_{30}$ $L_{21}$ $L_{12}$ $L_{03}$ | third degree |
| $L_{40}$ $L_{31}$ $L_{22}$ $L_{13}$ $L_{04}$ | fourth degree |
| $L_{50}$ $L_{41}$ $L_{32}$ $L_{23}$ $L_{14}$ $L_{05}$ | fifth degree |
| . | . |
| . | . |
| . | . |

Figure 2.1

Accordingly, for an approximation retaining up to and including all third degree tensor terms, nine individual $L_{jk}$ need be identified; for fourth degree terms, the number would be fourteen, and so on.

In the remaining sections of this chapter the identification scheme is described and illustrated. However, a minor point should be mentioned here before proceeding further. When the $L_{jk}$ are identified, the pre-

ceding constant term associated with each--that is, 1/j!k!--is suppressed into the operator itself. This obviously will cause no loss of generality, since any one of the constants may be extracted (if so desired) from the corresponding operator by a simple matrix scaling operation. Furthermore, as will be seen in the next section, the operators that are identified are of reduced size as compared to those used in the discussion to this point.

### 2.1.2 Software Adaptation

The major difference between the algorithm here and the one employed in the first and second order linearizations of the previous two chapters is, of course, the necessary addition of control variables. However, a change more fundamental to the basic technique is the need now for a greater number of time points in the sampling process. This is basically due to the behavior of the input forcing function, taken to be sinusoid for the examples treated in following sections. Obviously, the sampling period must be somewhat smaller than the period of oscillation of the input. For ease in programming, a constant sampling rate is used. Another major adaptation in the software is the development of a scheme for logical ordering of the terms which arise from the tensor products.

Consider again a series expansion, given now in terms of the vectors $x$ and $u$,

$$\dot{x} = L_{10}\, x + L_{01}\, u + L_{20}\, x \bullet x + L_{11}\, x \bullet u + L_{02}\, u \bullet u + L_{30}\, x \bullet x \bullet x + \ldots$$

Since the task is to identify the parameters of the $L_{jk}$ only, the new vector equation

$$\dot{x} = [L_{10} \vdots L_{01} \vdots L_{20} \vdots L_{11} \vdots L_{02} \vdots L_{30} \vdots \ldots] \begin{bmatrix} x \\ \cdots\cdots \\ u \\ \cdots\cdots \\ x \otimes x \\ \cdots\cdots \\ x \otimes u \\ \cdots\cdots \\ u \otimes u \\ \cdots\cdots \\ x \otimes x \otimes x \\ \cdots\cdots \\ \vdots \end{bmatrix}$$

would be constructed.

Recall that for the case of the homogeneous system several redundant terms appeared in the vector stacked with tensor product terms. In such a case it was argued that those terms, with the corresponding columns of the identified operator, could be eliminated for purposes of the identification. Such is the case here; for instance, consider the product term

$$x \otimes x \otimes u$$

which corresponds to the $L_{21}$ operator. For three states and two inputs, this product can be constructed according to the previous scheme:

$$x \otimes x \otimes u = x \otimes (x \otimes u)$$

$$= x \otimes \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} [u_1 \; u_2]$$

$$= x \otimes \begin{bmatrix} x_1u_1 & x_1u_2 \\ x_2u_1 & x_2u_2 \\ x_3u_1 & x_3u_2 \end{bmatrix} .$$

Again, the term $(x \otimes u)$ can be considered as a six-dimensional object, and stacked in one (row) vector. Thus,

$$x \otimes x \otimes u = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \begin{bmatrix} x_1u_1 & x_1u_2 & x_2u_1 & x_2u_2 & x_3u_1 & x_3u_2 \end{bmatrix}$$

$$= \begin{bmatrix} x_1^2u_1 & x_1^2u_2 & x_1x_2u_1 & x_1x_2u_2 & x_1x_3u_1 & x_1x_3u_2 \\ x_2x_1u_1 & x_2x_1u_2 & x_2^2u_1 & x_2^2u_2 & x_2x_3u_1 & x_2x_3u_2 \\ x_3x_1u_1 & x_3x_1u_2 & x_3x_2u_1 & x_3x_2u_2 & x_3^2u_1 & x_3^2u_2 \end{bmatrix},$$

an 18-dimensional object. Therefore, the linear operator $L_{21}$ associated with this product term would have dimension $3 \times 18$. But notice that there are six identical terms in the product:

$$x_1x_2u_1 = x_2x_1u_1,$$
$$x_1x_2u_2 = x_2x_1u_2,$$
$$x_1x_3u_1 = x_3x_1u_1,$$
$$x_1x_3u_2 = x_3x_1u_2,$$
$$x_2x_3u_1 = x_3x_2u_1,$$
$$x_2x_3u_2 = x_3x_2u_2.$$

Eliminating the redundant terms of $x \otimes x \otimes u$ would leave a vector of length 12. Likewise, the corresponding operator would have dimension $3 \times 12$; denote this matrix as $\tilde{L}_{21}$.

Now a problem of reduced size can be formulated. The equation

$$\dot{x} = [\tilde{L}_{10} \vdots \tilde{L}_{01} \vdots \tilde{L}_{20} \vdots \tilde{L}_{11} \vdots \tilde{L}_{02} \vdots \tilde{L}_{30} \vdots \ldots]\, x_L$$

is constructed, where the $\tilde{L}_{jk}$ correspond to the reduced number of tensor

product terms stacked in $x_L$. It will not always be the case that $\tilde{L}_{jk} \neq L_{jk}$; in fact, $\tilde{L}_{10} = L_{10}$ and $\tilde{L}_{01} = L_{01}$. But the new notation will be employed for consistency. The bracketed term in the above equation has dimension $n \times p$, with $n$ being the number of independent state variables and $p$ being dependent upon the number of tensor terms retained in the approximation. Therefore, $\dot{x}$ has dimension $n \times 1$ and $x_L$ had dimension $p \times 1$. The composition of the vector $x_L$ is a crucial consideration in the equation; a logical scheme is necessary for consistency, presented in the following.

Let $S$ be the space of states so that the n-vector $x$ is an element of $S$. Let $U$ be the space of inputs so that the m-vector $u$ is an element of $U$. Furthermore, let $\{a_1, a_2, \ldots, a_n\}$ be a basis for the space $S$ and $\{b_1, b_2, \ldots, b_m\}$ be a basis for the space $U$. The space

$$\underbrace{S \otimes S \otimes \ldots \otimes S}_{q \text{ times}}$$

would have a basis of q-vectors, $n^q$ in number, of the form

$$a_{i_1} \otimes a_{i_2} \otimes \ldots \otimes a_{i_q},$$

where $i_1, i_2, \ldots, i_q$ are integers between 1 and n. Such a space would have linear combinations of products,

$$x^1 \otimes x^2 \otimes \ldots \otimes x^q,$$

of $q$ n-vectors, each comprising a $n^q$-dimensional object. But, as seen in the discussion above, this number can be reduced if redundant terms within the product are eliminated. The number of distinct elements in the product is given by

$$\binom{n + q - 1}{q},$$

that is, "The combination of $(n + q - 1)$ items taken $q$ at a time." Likewise, to span the space

$$\underbrace{U \otimes U \otimes \ldots \otimes U}_{r \text{ times}}$$

$m^r$ r-vectors of the form

$$b_{j_1} \otimes b_{j_2} \otimes \ldots \otimes b_{j_r},$$

where $j_1, j_2, \ldots, j_r$ are integers between 1 and m, would be necessary. But the number of distinct elements in a product

$$u^1 \otimes u^2 \otimes \ldots \otimes u^r$$

would be given by

$$\binom{m + r - 1}{r}.$$

The extension of this to product spaces of the form

$$\underbrace{S \otimes S \otimes \ldots \otimes S}_{q \text{ times}} \otimes \underbrace{U \otimes U \otimes \ldots \otimes U}_{r \text{ times}}$$

follows naturally. The number of $(q+r)$-vectors of the form

$$a_{i_1} \otimes \ldots \otimes a_{i_q} \otimes b_{j_1} \otimes \ldots \otimes b_{j_r}$$

in the basis is given by the product of $n^q$ and $m^r$. Here, the number of distinct elements in a product

$$x^1 \otimes \ldots \otimes x^q \otimes u^1 \otimes \ldots \otimes u^r$$

is calculated according to

$$\binom{n + q - 1}{q} \cdot \binom{m + r - 1}{r}.$$

Knowing the number of elements in $x_L$ is half the problem; some ordering convention need be established. To settle this issue, consider a simple illustration with $n = 3$ and $m = 2$; thus, $\{a_1, a_2, a_3\}$ forms a

basis for $S$ and $\{b_1, b_2\}$ forms a basis for $U$. Then, for the product space

$$S \otimes S \otimes U ,$$

a legitimate basis would be

$$\{a_1 \otimes a_1 \otimes b_1,\ a_1 \otimes a_1 \otimes b_2,\ a_1 \otimes a_2 \otimes b_1,\ a_1 \otimes a_2 \otimes b_2,$$
$$a_1 \otimes a_3 \otimes b_1,\ a_1 \otimes a_3 \otimes b_2,\ a_2 \otimes a_1 \otimes b_1,\ a_2 \otimes a_1 \otimes b_2,$$
$$a_2 \otimes a_2 \otimes b_1,\ a_2 \otimes a_2 \otimes b_2,\ a_2 \otimes a_3 \otimes b_1,\ a_2 \otimes a_3 \otimes b_2,$$
$$a_3 \otimes a_1 \otimes b_1,\ a_3 \otimes a_1 \otimes b_2,\ a_3 \otimes a_2 \otimes b_1,\ a_3 \otimes a_2 \otimes b_2,$$
$$a_3 \otimes a_3 \otimes b_1,\ a_3 \otimes a_3 \otimes b_2\} ,$$

consisting of 18, or $(3^2) \cdot (2^1)$, members. The space consists of linear combinations of products of the form

$$x \otimes x \otimes u$$

which would also consist of 18 elements; it is the ordering of these elements which is the primary concern here.

Note the manner in which the members of the above-mentioned basis are listed. By this convention, the index of each item in the general form

$$a_{i_1} \otimes a_{i_2} \otimes b_{j_1}$$

is incremented through, beginning at the right. For instance, with $i_1 = 1$, $i_2 = 1$ initially, $j_1$ is incremented from 1 to r, at which time the index $i_2$ is incremented by one and $j_1$ is incremented from 1 to r again. This process is continued for $i_2 = 1,2,\ldots,q$, at which time $i_1$ increments by one and the previous steps are repeated. Thus, with $i_1 = 1,2,\ldots,q$, the last basis element in the list will always be $a_q \otimes a_q \otimes b_r$. The $(n^q) \cdot (m^r)$ elements of the tensor product $x \otimes x \otimes u$ can be listed in a similar fashion, the obvious difference being that these elements are ordinary products of the members of $x$ and $u$. To eliminate redundant terms

in the tensor products for identification of the corresponding $\hat{L}_{jk}$ requires a slight modification of this scheme.

The algorithm for ordering the non-redundant terms from the tensor products will now be presented via a flow diagram, represented in Figure 2.2. Variable names for this flow chart are detailed in the following:

X and U: arrays of N states and M inputs, respectively.
INDX: array of indexes for elements of x, ordered.
INDU: array of indexes for elements of u, ordered.
NXTR(i): (marker) cumulative number of elements in INDX for tensor products of x up to the i-th degree.
NUTR(i): (marker) cumulative number of elements in INDU for tensor products of u up to the i-th degree.
XPROD: array of products of combinations of elements of x.
UPROD: array of products of combinations of elements of u.
XPR(i): (marker) cumulative number of elements in XPROD for tensor products of x up to the i-th degree.
UPR(i): (marker) cumulative number of elements in UPROD for tensor products of u up to the i-th degree.
XL: array containing all non-redundant product terms of states with inputs, ordered.

Three basic divisions comprise the algorithm. The first, represented in Figure 2.2a, establishes the order of the indexes or subscripts of x and u to be used in forming products. This first stage is shown in a long form for clarity, since a general, condensed version would be less illustrative; certainly one set of nested loops would serve the general purpose as opposed to a set for each degree of tensor products as shown. To exemplify the ordering of these indexes, consider the previous case of the 3-vector x. For an approximation including third degree tensor product terms the array of indexes for x, ordered according to the left branch of Figure 2.2a, would be given by

INDX = (1 , 2 , 3 ; 1 1 , 1 2 , 1 3 , 2 2 , 2 3 , 3 3 ; 1 1 1 ,
1 1 2 , 1 1 3 , 1 2 2 , 1 2 3 , 1 3 3 , 2 2 2 , 2 2 3 ,

2 3 3 , 3 3 3) .

In the above, the notation is such that a semi-colon separates groups of indexes corresponding to the different degree of tensor product, and a comma separates those indexes which are to be used in ordering product combinations of $x_1$, $x_2$, and $x_3$. Thus, stage two (Figure 2.2b) uses these indexes to compute combinatorial product terms; the result for this case would be

$$\begin{aligned} \text{XPROD} = (&x_1, x_2, x_3; x_1x_1, x_1x_2, x_1x_3, x_2x_2, x_2x_3, x_3x_3; \\ &x_1x_1x_1, x_1x_1x_2, x_1x_1x_3, x_1x_2x_2, x_1x_2x_3, \\ &x_1x_3x_3, x_2x_2x_2, x_2x_2x_3, x_2x_3x_3, x_3x_3x_3) \ . \end{aligned}$$

Similar calculations are performed for the m-vector u (as presented, the algorithm assumes $m > 0$), so that in the final stage, Figure 2.2c, the product terms are combined and ordered in the vector $x_L$, corresponding to the $\tilde{L}_{jk}$ order chosen.

These final two stages of the algorithm, as represented in the flowchart, are given for the general case (arbitrary degree of approximation). The construction is such that the number of multiplications performed is held to a minimum. Furthermore, the division of the algorithm into three stages offers some conceptual convenience as well. It is not difficult to see from this a method for implementation on a digital computer.

The heart of the identification scheme, then, is embodied in the construction of the p-vector $x_L$. The following example will serve to summarize the principle.

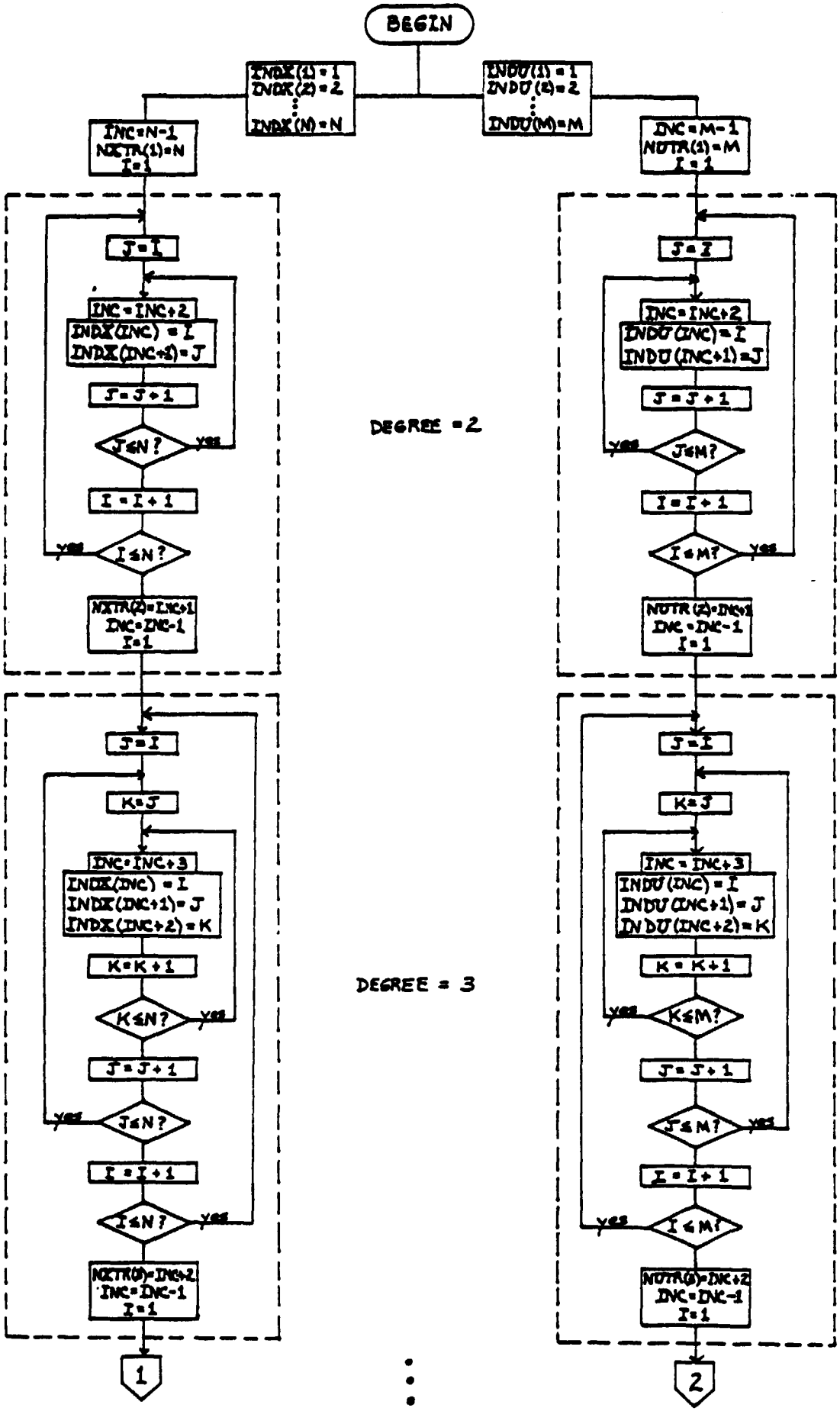

BEGIN
INDX(1) = 1
INDX(2) = 2
INDX(N) = N
INDU(1) = 1
INDU(2) = 2
INDU(M) = M
INC = N-1
NXTR(1) = N
I = 1
INC = M-1
NUTR(1) = M
I = 1
J = I
INC = INC+2
INDX(INC) = I
INDX(INC+1) = J
J = J+1
J≤N?
yes
I = I+1
I≤N?
yes
NXTR(2) = INC+1
INC = INC-1
I = 1
INDU(INC) = I
INDU(INC+1) = J
J≤M?
I≤M?
NUTR(2) = INC+1
K = J
INC = INC+3
INDX(INC) = I
INDX(INC+1) = J
INDX(INC+2) = K
K = K+1
K≤N?
J≤N?
NXTR(3) = INC+2
INDU(INC) = I
INDU(INC+1) = J
INDU(INC+2) = K
K≤M?
J≤M?
NUTR(3) = INC+2
DEGREE = 2
DEGREE = 3
1
2

Figure 2.2a

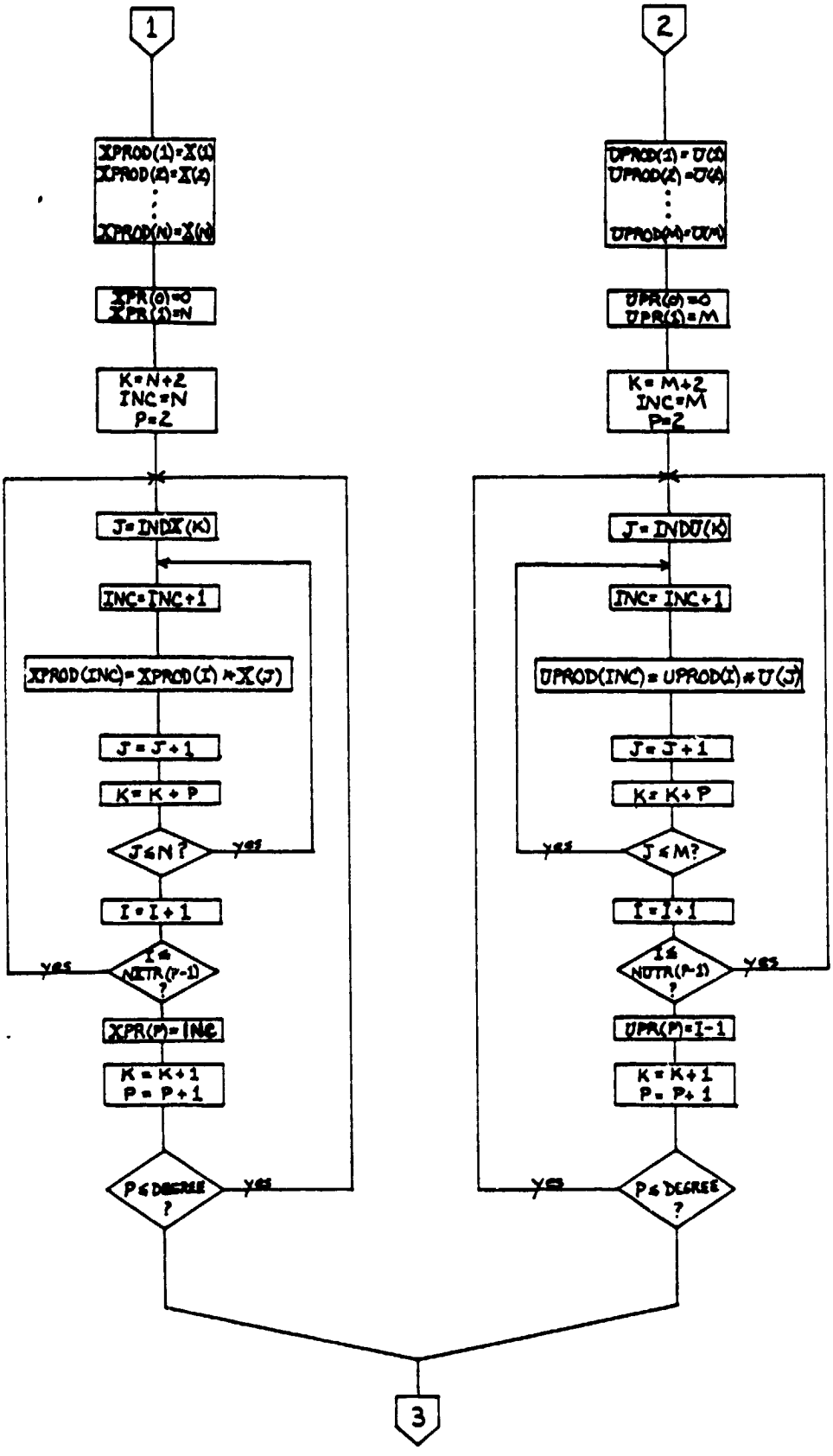

1
XPROD(1)=X(1)
XPROD(2)=X(2)
XPROD(N)=X(N)
XPR(0)=0
XPR(1)=N
K=N+2
INC=N
P=2
J=INDX(K)
INC=INC+1
XPROD(INC)=XPROD(I)*X(J)
J=J+1
K=K+P
J≤N?
yes
I=I+1
I≤ NUTR(P-1) ?
yes
XPR(P)=INC
K=K+1
P=P+1
P≤DEGREE ?
yes
2
UPROD(1)=U(1)
UPROD(2)=U(2)
UPROD(M)=U(M)
UPR(0)=0
UPR(1)=M
K=M+2
INC=M
P=2
J=INDU(K)
INC=INC+1
UPROD(INC)=UPROD(I)*U(J)
J=J+1
K=K+P
yes
J≤M?
I=I+1
I≤ NUTR(P-1) ?
yes
UPR(P)=I-1
K=K+1
P=P+1
yes
P≤DEGREE ?
3

Figure 2.2b

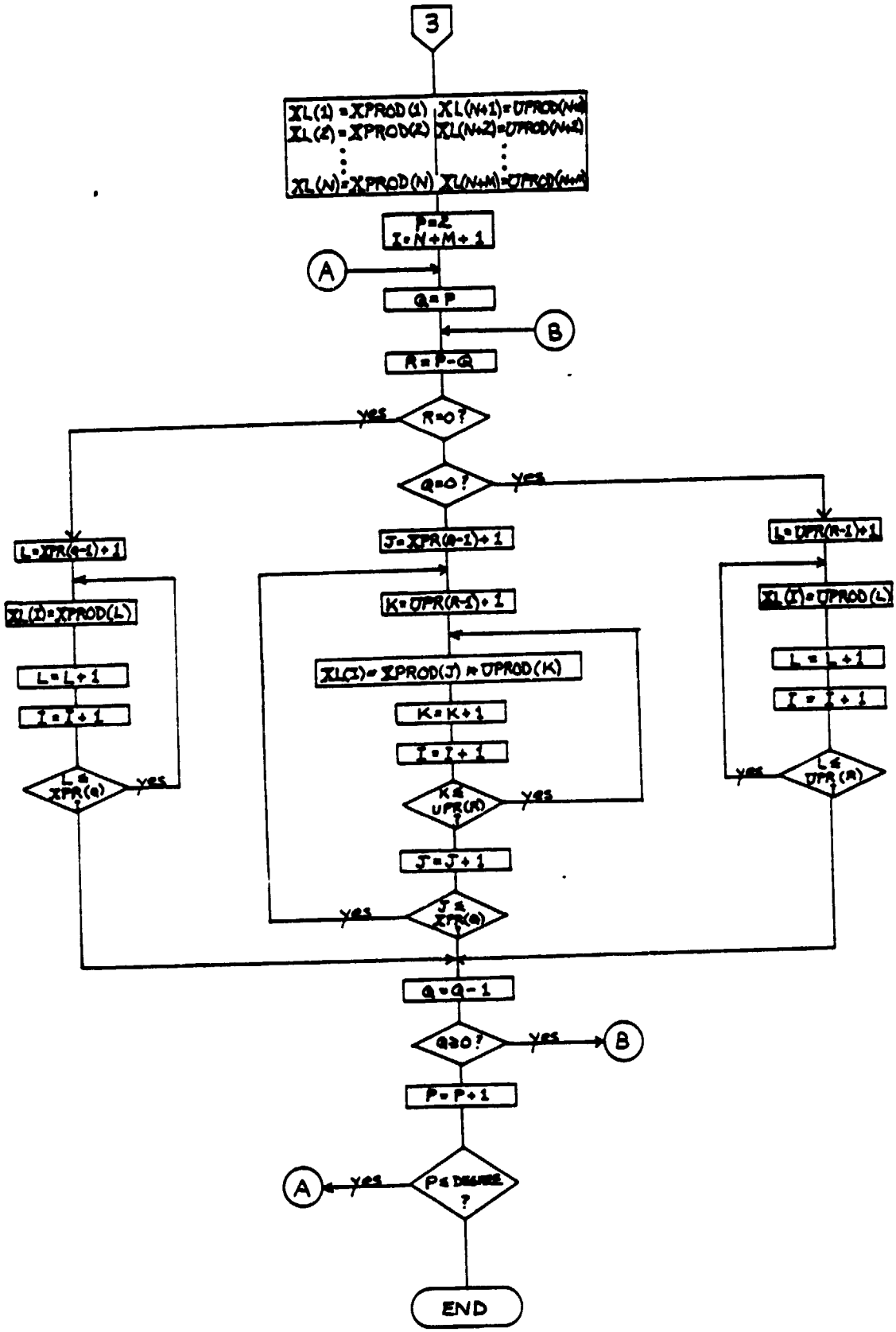

3
XL(1) = XPROD(1) XL(N+1) = UPROD(N+1)
XL(2) = XPROD(2) XL(N+2) = UPROD(N+2)
XL(N) = XPROD(N) XL(N+M) = UPROD(N+M)
P=2
I=N+M+1
A
Q=P
B
R=P-Q
R=0?
yes
Q=0?
yes
L=XPR(Q-1)+1
XL(I)=XPROD(L)
L=L+1
I=I+1
L ≤ XPR(Q) ?
yes
J=XPR(Q-1)+1
K=UPR(R-1)+1
XL(I)=XPROD(J)*UPROD(K)
K=K+1
I=I+1
K ≤ UPR(R) ?
yes
J=J+1
J ≤ XPR(Q) ?
yes
L=UPR(R-1)+1
XL(I)=UPROD(L)
L=L+1
I=I+1
L ≤ UPR(R) ?
yes
Q=Q-1
Q≥0?
yes
B
P=P+1
P ≤ DEGREE ?
yes
A
END

Figure 2.2c

Consider the nonhomogeneous vector differential equation

$$\dot{x} = f(x,u),$$

with some initial condition $x(0) = x_0$. Let the state and input vectors be given by

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \; ; \quad u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}.$$

Suppose that in the approximation up to (and including) third degree tensor terms are kept. Thus, the expansion, with tensor notation applied, is given by

$$\begin{aligned} f(x,u) \tilde{=} & \; L_{10}\, x + L_{01}\, u \\ & + L_{20}\, x \circledcirc x + L_{11}\, x \circledcirc u \\ & + L_{02}\, u \circledcirc u + L_{30}\, x \circledcirc x \circledcirc x \\ & + L_{21}\, x \circledcirc x \circledcirc u + L_{12}\, x \circledcirc u \circledcirc u \\ & + L_{03}\, u \circledcirc u \circledcirc u \; . \end{aligned}$$

Putting this into the form discussed above gives

$$\dot{x} = [\tilde{L}_{10}\ \tilde{L}_{01}\ \tilde{L}_{20}\ \tilde{L}_{11}\ \tilde{L}_{02}\ \tilde{L}_{30}\ \tilde{L}_{21}\ \tilde{L}_{12}\ \tilde{L}_{03}]\, x_L \; .$$

The number of elements in $x_L$ is calculated according to

$$p = \sum_{i=1}^{9} p_i \; ,$$

where each $p_i$ corresponds to one of the linear operators. Then,

$$p = \binom{3+1-1}{1} + \binom{2+1-1}{1} + \binom{3+2-1}{2}$$

$$
\begin{aligned}
&+ \binom{3+1-1}{1} \cdot \binom{2+1-1}{1} + \binom{2+2-1}{2} \\
&+ \binom{3+3-1}{3} + \binom{3+2-1}{2} \cdot \binom{2+1-1}{1} \\
&+ \binom{3+1-1}{1} \cdot \binom{2+2-1}{2} + \binom{2+3-1}{3} \\
&= 3 + 2 + 6 \\
&+ 3 \cdot 2 + 3 \\
&+ 10 + 6 \cdot 2 \\
&+ 3 \cdot 3 + 4 \\
&= 55.
\end{aligned}
$$

Rather than list all of the 55 members here, consider just a few of the partitions to be stacked in the vector $x_L$:

$$
\text{from } \underset{\sim}{x} \otimes \underset{\sim}{x}: \begin{bmatrix} x_1x_1 \\ x_1x_2 \\ x_1x_3 \\ x_2x_2 \\ x_2x_3 \\ x_3x_3 \end{bmatrix} \qquad \text{from } \underset{\sim}{x} \otimes \underset{\sim}{u}: \begin{bmatrix} x_1u_1 \\ x_1u_2 \\ x_2u_1 \\ x_2u_2 \\ x_3u_1 \\ x_3u_2 \end{bmatrix}
$$

$$
\text{from } \underset{\sim}{x} \otimes \underset{\sim}{x} \otimes \underset{\sim}{u}: \begin{bmatrix} x_1x_1u_1 \\ x_1x_1u_2 \\ x_1x_2u_1 \\ x_1x_2u_2 \\ x_1x_3u_1 \\ x_1x_3u_2 \\ x_2x_2u_1 \\ x_2x_2u_2 \\ x_2x_3u_1 \\ x_2x_3u_2 \\ x_3x_3u_1 \\ x_3x_3u_2 \end{bmatrix} \qquad \text{from } \underset{\sim}{x} \otimes \underset{\sim}{u} \otimes \underset{\sim}{u}: \begin{bmatrix} x_1u_1u_1 \\ x_1u_1u_2 \\ x_1u_2u_2 \\ x_2u_1u_1 \\ x_2u_1u_2 \\ x_2u_2u_2 \\ x_3u_1u_1 \\ x_3u_1u_2 \\ x_3u_2u_2 \end{bmatrix}
$$

This illustration should further establish the ordering convention.

With the construction of the equation complete, the business of identification of the $\hat{\gamma}_{jk}$ may be undertaken. The n simultaneous nonlinear differential equations are integrated and the resulting data is sampled at the selected h time points. These sampled values are loaded into the matrix $X_L$, which is now pxh in size. Note that the first n+m rows of $X_L$ are determined from the sampled values of x and u; the remaining p-(n+m) rows are multiples and combinations of those first n+m rows. Finally, the $\dot{X}$ matrix, dimension nxh, is formed by loading derivative estimates for $\dot{x}_1, \dot{x}_2, \ldots, \dot{x}_n$ at the h time points.

With this, the matrix equation assumes the form

$$\underset{\text{nxh}}{\dot{X}} = \underset{\text{nxp}}{[\hat{\gamma}_{10}\ \hat{\gamma}_{01}\ \hat{\gamma}_{20}\ \hat{\gamma}_{11}\ \hat{\gamma}_{02}\ \hat{\gamma}_{30}\ \hat{\gamma}_{21}\ \hat{\gamma}_{12}\ \hat{\gamma}_{03}]}\ \underset{\text{pxh}}{X_L},$$

with matrix dimensions as shown. The least squares minimization algorithm is again used in identifying the nxp partitioned matrix containing the desired $\hat{\gamma}_{jk}$. The matrix returned will have five partitions if only second degree terms are kept, nine partitions if third degree terms are added, and so on. The sorting of the partitions is merely a numerical bookkeeping job, since their sizes are known in advance.

As noted at the beginning of this section, more time points are required for sampling in the identification now, due to the behavior of the forcing functions. Whereas previously 15 sampling times were used, not all evenly spaced, now h=40 points are used with a constant sampling rate. The number of points and the rate may be varied according to the

demands of the problem.

The next section illustrates the application of the software on several representative examples, each one progressing in complexity. In choosing the nonhomogeneous vector differential equation for an example, two conditions must be met:

1) $L_{10}$ must be stable at the origin, that is, have eigenvalues with negative real parts;

2) The origin must be an equilibrium point; that is,

$$f_i(0,0) = 0$$

for $i = 1,2,\ldots,n$.

Choice of an initial condition is not intuitively obvious in these nonlinear examples, so in all cases small pertubations from the origin are chosen as initial conditions.

### 2.1.3 Second Order Examples

This section treats two examples, each of which involves a state vector of two elements and an input vector of two elements. Keeping up to third degree tensor terms, the nonhomogeneous, nonlinear vector differential equation

$$\dot{x} = f(x,u)$$

is approximated by

$$\dot{x} \tilde{=} L_{10}\, x + L_{01}\, u + L_{20}\, x \circledcirc x + L_{11}\, x \circledcirc u + L_{02}\, u \circledcirc u$$

$$+ L_{30}\, x \otimes x \otimes x + L_{21}\, x \otimes x \otimes u$$
$$+ L_{12}\, x \otimes u \otimes u + L_{03}\, u \otimes u \otimes u \ .$$

In each case, for a given operating point $(x_0,u_0)$, the nine individual linear operators are identified, and then used to reformulate the two equation system to be integrated for comparison to the true solution of the original equation.

For the first example, consider the system of two exact nonhomogeneous differential equations

$$f_1(x,u) = \dot{x}_1$$
$$= x_1^2 u_1 + u_2^2 - x_1 \ ,$$

$$f_2(x,u) = \dot{x}_2$$
$$= x_1 x_2 + u_1^2 u_2 - 2x_2 \ ,$$

where the state vector is

$$x = (x_1,x_2)' \ .$$

and the input vector

$$u = (u_1,u_2)' \ .$$

The initial conditions to be used in the identification of the $\tilde{L}_{jk}$ are given by

$$x_1(0) = 0.2 \ ,$$
$$x_2(0) = 0.4 \ .$$

Notice that the condition of stability is met, that is

$$L_{10} = \left[\begin{matrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{matrix}\right]\Bigg|_{\substack{x_1 = 0,\ x_2 = 0 \\ u_1 = 0,\ u_2 = 0}}$$

$$= \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix}$$

has eigenvalues with negative real parts. Furthermore,

$$f_i(0, 0) = 0$$

for $i = 1,2$, that is, the origin is an equilibrium point.

The input forcing functions chosen are sinusoidal in nature, and are given by

$$u_1(t) = \sin(2\pi f_1 t) ,$$

$$u_2(t) = (0.5) \sin(2\pi f_2 t) ,$$

for $f_1 = 5$ hertz and $f_2 = 10$ hertz.

A fourth-order Runge-Kutta routine is employed to integrate the vector differential equation; the same routine (and integration stepsize) is employed later to integrate the system as embodied in the identified model. Thus, a block of data---the "true solution"---is established and stored in arrays corresponding to the $x_1$ and $x_2$ solutions. The stepsize in the integration is taken to be 0.005 so as to comprise ample information of function behavior for the derivative estimation. Note that for a frequency of 10 hertz for $u_2$, corresponding to a period

of 0.10, each cycle is visited 20 times in the integration.

With the data tabulated, the sampling operation ensues. The true solution is sampled at 40 points in time at a constant sampling rate of 40 samples per second, corresponding to a sample period of 0.025.

Thus, for the matrix equation

$$\dot{X} = [\tilde{L}_{10}\ \tilde{L}_{01}\ \tilde{L}_{20}\ \tilde{L}_{11}\ \tilde{L}_{02}\ \tilde{L}_{30}\ \tilde{L}_{21}\ \tilde{L}_{12}\ \tilde{L}_{03}]\ X_L\ ,$$

the matrix $X_L$ has 34 rows and 40 columns. As explained in the previous section, the number of tensor product terms (rows of $X_L$) is calculated using the general formula

$$p_i = \binom{n+q-1}{q} \cdot \binom{m+r-1}{r}$$

for the $q$ occurrences of $x$ and $r$ occurrences of $u$ in the $i$-th product. Then the total number of rows of $X_L$ is found according to

$$p = \sum_{i=1}^{9} p_i\ ,$$

and, for this particular example,

$$p = 2 + 2 + 3 + 4 + 3 + 4 + 6 + 6 + 4$$
$$= 34.$$

The 40 columns of $\dot{X}$ and $X_L$ correspond, of course, to the number of time points used in the sampling.

The data for the $\dot{X}$ matrix (dimension 2 x 40) is obtained directly from the (stored) true solution, where the derivatives are estimated according to

$$[\dot{X}]_{ij} = \frac{x_i(t_j+0.005) - x_i(t_j-0.005)}{2(0.005)}$$

for $i = 1,2$ and $j = 1,2,\ldots,40$, where the $t_j$ are the points in time

chosen for sampling and $[\dot{X}]_{ij}$ represents the ij-th element of $\dot{X}$.

Having completed the sampling, derivative estimation, and loading, the least squares minimization algorithm (routine SIMEQUAT) is executed on the matrix equation. Knowing the sizes $(2xp_i)$ of the individual linear operators facilitates a partitioning of the 2x34 matrix returned in the identification scheme. In Figure 2.3 these operators are shown with the corresponding partition of the $x_L$ vector. The eigenvalues of $\tilde{L}_{10}$ are both negative and real. It is interesting to attach some meaning to a few of these numbers in relation to the original equations. For example, note the relative size of the (2,2) element of $\tilde{L}_{20}$, corresponding to the presence of term $x_1x_2$ in the second equation; the relative magnitude of the (1,3) element of $\tilde{L}_{02}$ indicates the occurrence of the $u_2^2$ term of the first equation; the relative magnitude of the (2,2) element of $\tilde{L}_{03}$ represents the occurrence of the term $u_1^2u_2$ in the second equation.

The next step is to integrate the system with the coefficients as identified in the third degree approximation. To do this, the equation

$$\dot{x} = [\tilde{L}_{10}\ \tilde{L}_{01}\ \tilde{L}_{20}\ \tilde{L}_{11}\ \tilde{L}_{02}\ \tilde{L}_{30}\ \tilde{L}_{21}\ \tilde{L}_{12}\ \tilde{L}_{03}]\ x_L$$

is reconstructed so that the system of two differential equations, each with 34 terms in the sum, may be solved yielding x. In the first analysis, the same initial conditions used in the identification of the $\tilde{L}_{jk}$ are employed to simulate the original system, with the same forcing functions as used in the identification. The results of this test are depicted in Figures 2.4 and 2.5. As will be the case in all plots to follow, curve A represents the "true solution" of the original system as given by the fourth-order Runge-Kutta algorithm; curve B represents the

| LINEAR OPERATOR | PARTITION OF $x_L$ |
|---|---|
| $\mathcal{L}_{10} = \begin{bmatrix} -0.947 & -0.234 \\ 0.255 & -2.490 \end{bmatrix}$ | $(x_1, x_2)'$ |
| $\mathcal{L}_{01} = \begin{bmatrix} -0.007 & -0.026 \\ 0.041 & 0.071 \end{bmatrix}$ | $(u_1, u_2)'$ |
| $\mathcal{L}_{20} = \begin{bmatrix} 0.406 & 2.688 & -0.298 \\ -3.468 & 8.144 & -0.965 \end{bmatrix}$ | $(x_1^2, x_1x_2, x_2^2)'$ |
| $\mathcal{L}_{11} = \begin{bmatrix} 0.122 & 0.352 & 0.000 & -0.052 \\ -0.547 & -06.26 & 0.097 & 0.107 \end{bmatrix}$ | $(x_1u_1, x_1u_2, x_2u_1, x_2u_2)'$ |
| $\mathcal{L}_{02} = \begin{bmatrix} 0.000 & 0.000 & 0.935 \\ 0.001 & 0.001 & -0.004 \end{bmatrix}$ | $(u_1^2, u_1u_2, u_2^2)'$ |
| $\mathcal{L}_{30} = \begin{bmatrix} -2.635 & -7.562 & 1.737 & -0.093 \\ 11.79 & -25.92 & 6.767 & -0.466 \end{bmatrix}$ | $(x_1^3, x_1^2x_2, x_1x_2^2, x_2^3)'$ |
| $\mathcal{L}_{21} = \begin{bmatrix} 0.505 & -1.203 & 0.041 & 0.351 & -0.003 & -0.025 \\ 1.831 & 2.117 & -0.640 & -0.715 & 0.049 & 0.055 \end{bmatrix}$ | $(x_1^2u_1, x_1^2u_2, x_1x_2u_1, x_1x_2u_2, x_2^2u_1, x_2^2u_2)'$ |
| $\mathcal{L}_{12} = \begin{bmatrix} 0.000 & -0.003 & 0.001 & -0.000 & 0.001 & -0.000 \\ -0.005 & -0.005 & 0.018 & 0.001 & 0.001 & -0.003 \end{bmatrix}$ | $(x_1u_1^2, x_1u_1u_2, x_1u_2^2, x_2u_1^2, x_2u_1u_2, x_2u_2^2)'$ |
| $\mathcal{L}_{03} = \begin{bmatrix} -0.000 & 0.000 & -0.000 & -0.000 \\ -0.000 & 0.935 & -0.001 & -0.002 \end{bmatrix}$ | $(u_1^3, u_1^2u_2, u_1u_2^2, u_2^3)'$ |

Figure 2.3

simulated solution which again employs the Runge-Kutta routine to integrate the system as embodied in the identified $\tilde{L}_{jk}$. Figure 2.4 shows the transient region of solution $x_1$; the effects of the input functions are evidenced by the oscillations. The simulated solution tracks the true solution quite well in this region. Figure 2.5 shows a plot of $x_2$ from time 0.0 to time 0.5. Again, curve B overlays curve A. Since the model system was identified from samples taken out to one second (i.e., 40 time points spaced at 0.025), an obvious question would concern the performance of the model system for simulations beyond the transient region, out to four seconds. The plots for these simulations are given in Figures 2.6 and 2.7 for $x_1$ and $x_2$, respectively. These plots indicate that the model system simulates the original system well, far beyond the interval in which samples were taken in the identification procedure. As might be expected by inspection of the original equations, after transients have settled the forcing functions tend to dominate the solution; this is depicted by the oscillatory motion about the origin for $x_2$ in Figure 2.7.

At this point it is interesting to discuss the concept of an "operating region", or feasibility region for initial conditions of the states of the system. For instance, the initial conditions chosen for this example were

$$x_1(0) = 0.2,$$
$$x_2(0) = 0.4,$$

and the nine linear operators for the series expansion were identified accordingly. But suppose that the model system as identified for the above-mentioned initial condition were used to simulate the original system at a different initial condition, namely

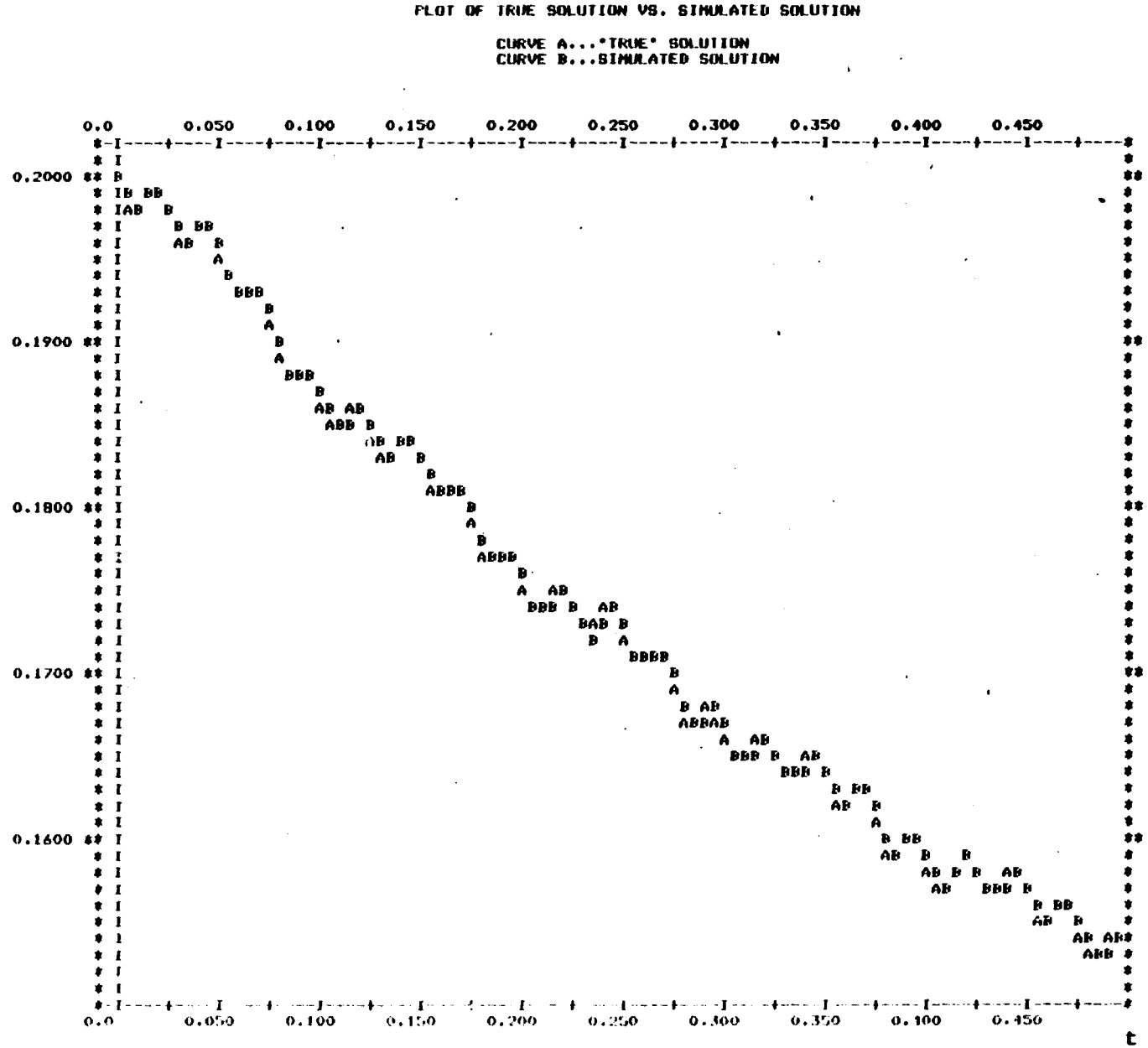

PLOT OF TRUE SOLUTION VS. SIMULATED SOLUTION
CURVE A...'TRUE' SOLUTION
CURVE B...SIMULATED SOLUTION
0.0
0.050
0.100
0.150
0.200
0.250
0.300
0.350
0.400
0.450
0.2000
0.1900
0.1800
0.1700
0.1600
x1
t

Figure 2.4

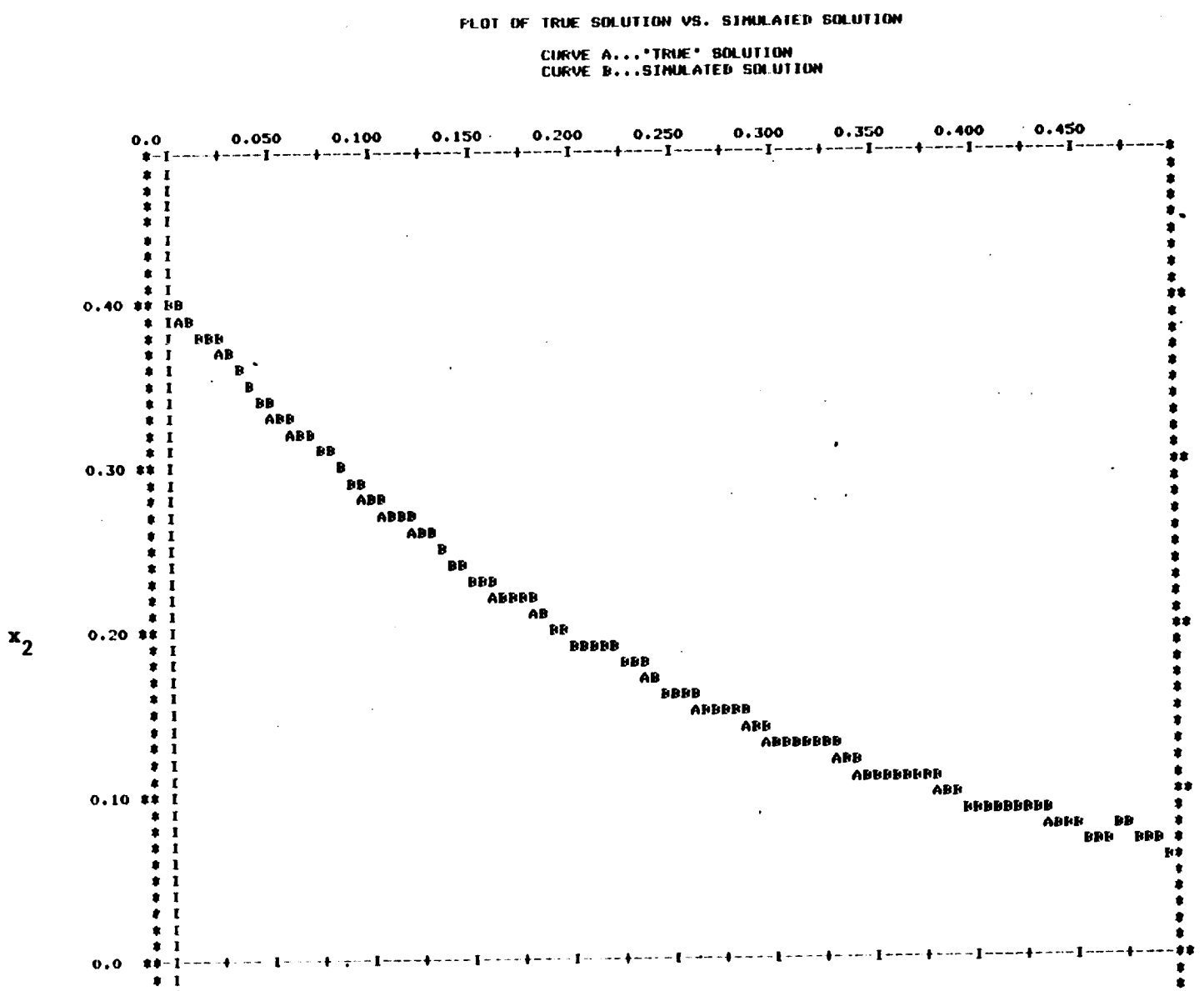

PLOT OF TRUE SOLUTION VS. SIMULATED SOLUTION
CURVE A...'TRUE' SOLUTION
CURVE B...SIMULATED SOLUTION
0.0
0.050
0.100
0.150
0.200
0.250
0.300
0.350
0.400
0.450
0.40
0.30
0.20
0.10
0.0
$x_2$
t

Figure 2.5

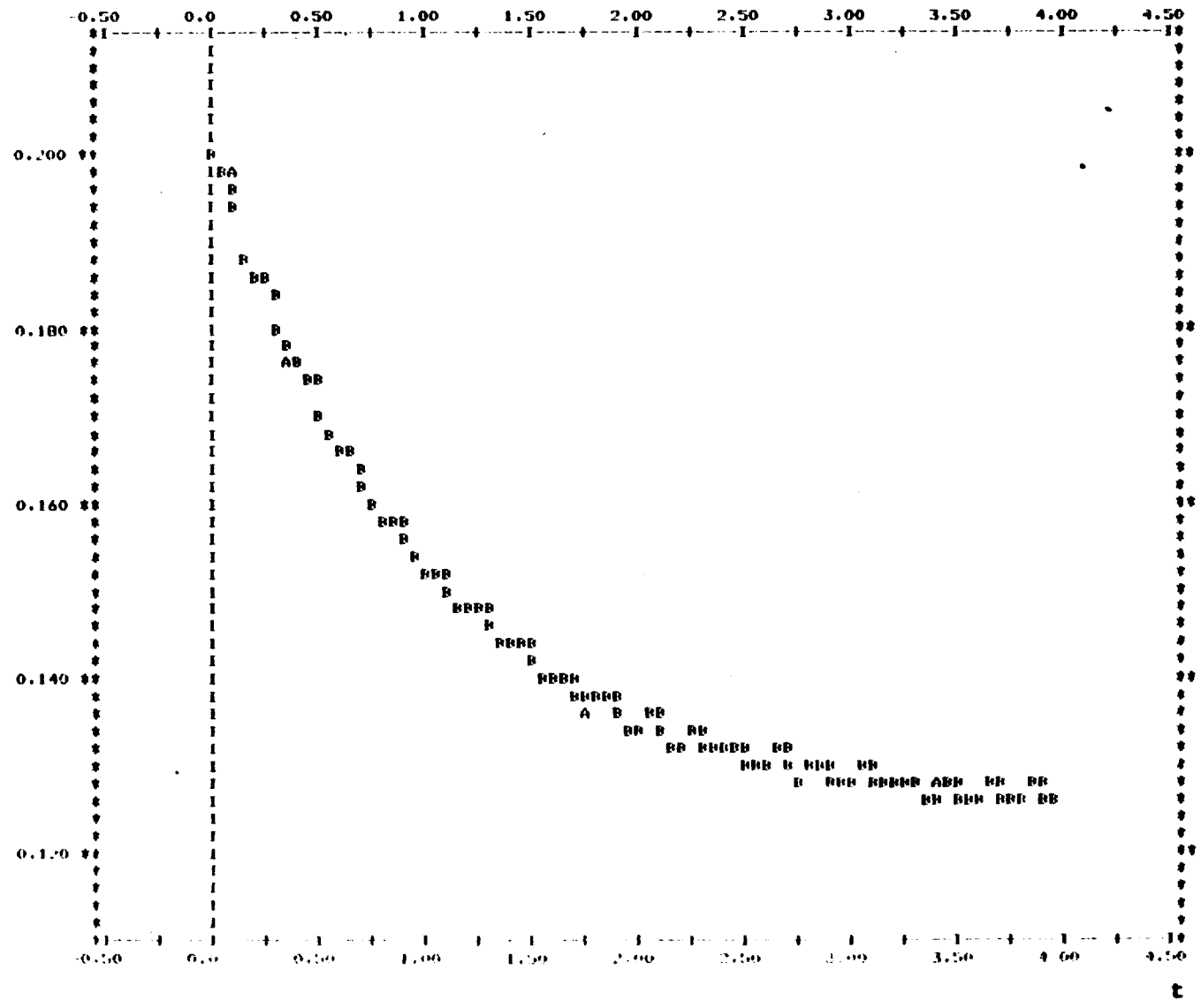



Figure 2.6

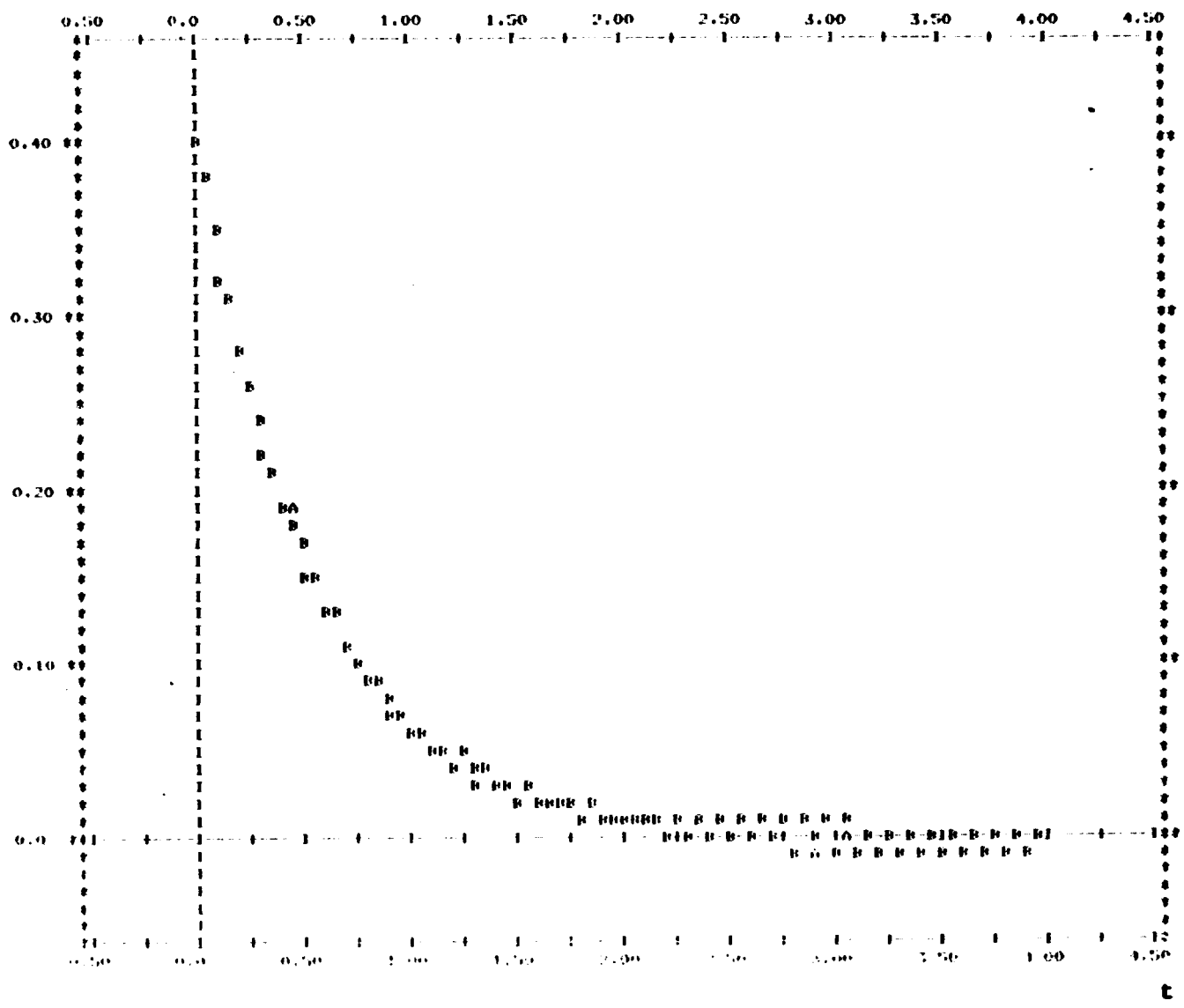

PLOT OF TRUE SOLUTION VS. SIMULATED SOLUTION
CURVE A...'TRUE' SOLUTION
CURVE B...SIMULATED SOLUTION
0.50
0.0
0.50
1.00
1.50
2.00
2.50
3.00
3.50
4.00
4.50
0.40
0.30
0.20
0.10
0.0
$x_2$
t

Figure 2.7

$$x_1(0) = 0.3,$$

$$x_2(0) = 0.3.$$

These initial conditions represent a change in magnitude of 50 and 25 percent, respectively, over the original initial conditions. The results of this test are given in Figures 2.8 and 2.9 for $x_1$ and $x_2$. A comparison of Figure 2.6 with Figure 2.8 shows a slight decline in tracking accuracy in the latter for the increased initial condition. But the plot for $x_2$ exhibits essentially identical accuracy as the simulation for the system as identified, since the initial condition was decreased in magnitude.

Before proceeding to discussion of the next example, an observation concerning model size is in order. Since there are only two terms of degree higher than two in the system, $x_1^2 u_1$ in the first question and $u_1^2 u_2$ in the second, one would suspect that an approximation keeping second degree tensor terms would suffice. Simulations for that identified system proved to be as accurate as the former approximation which kept up to third degree terms. Since a second degree approximation involves considerably less calculation (five partitions, 14 members in $x_L$). one might wish to consider using it in this case.

The second example in this section is more complex than the first in the sense that the nonhomogeneous system of differential equations is not exact, but a combination of hyperbolic functions. Again, intuition fails somewhat as to the behavior of the nonlinear functions, particularly in regard to choice of initial conditions.

Consider the following system:

$$\dot{x}_1 = f_1(x,u)$$

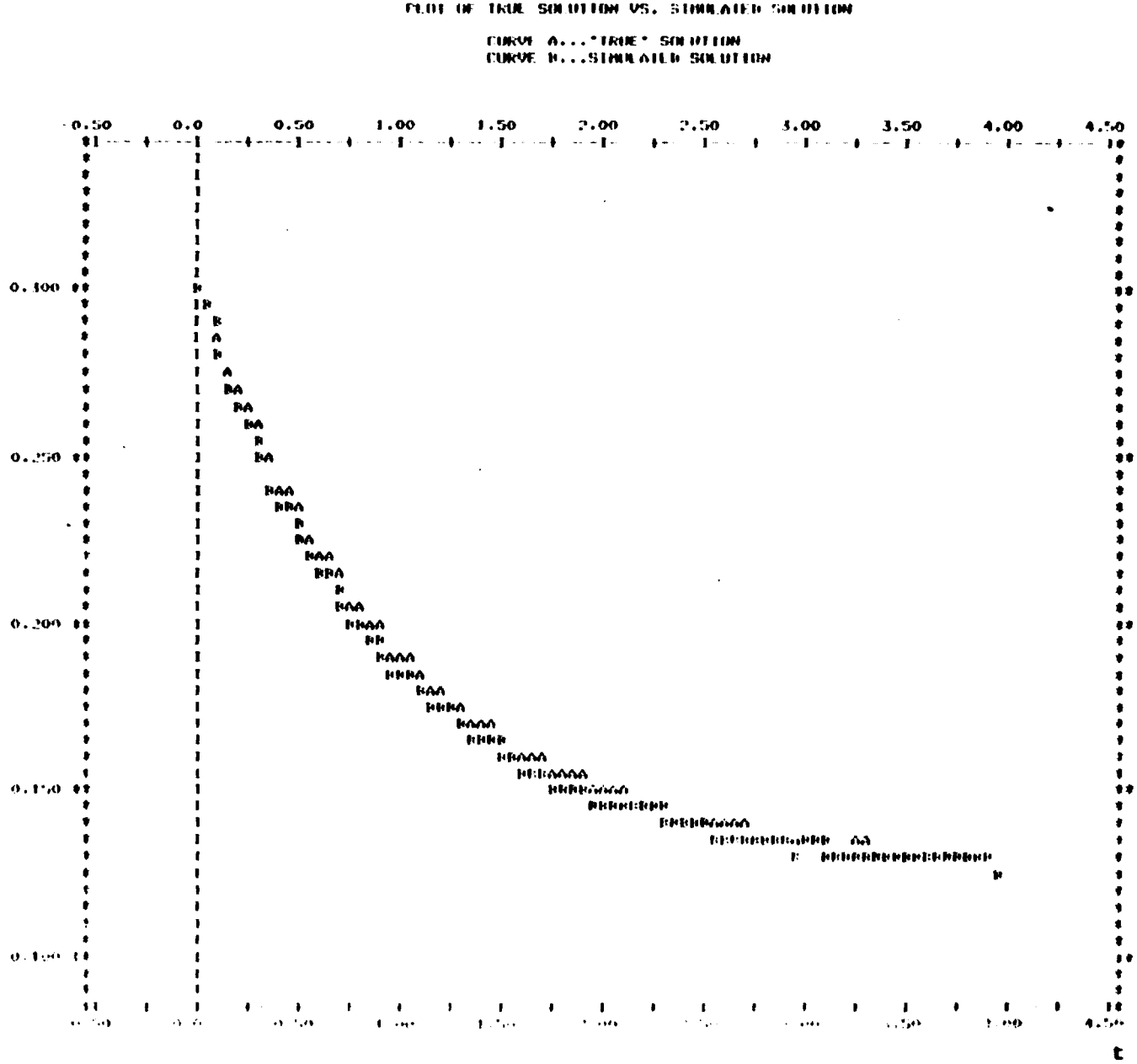

PLOT OF TRUE SOLUTION VS. SIMULATED SOLUTION
CURVE A...'TRUE' SOLUTION
CURVE B...SIMULATED SOLUTION
-0.50
0.0
0.50
1.00
1.50
2.00
2.50
3.00
3.50
4.00
4.50
0.300
0.250
0.200
0.150
0.100
$x_1$
t

Figure 2.8

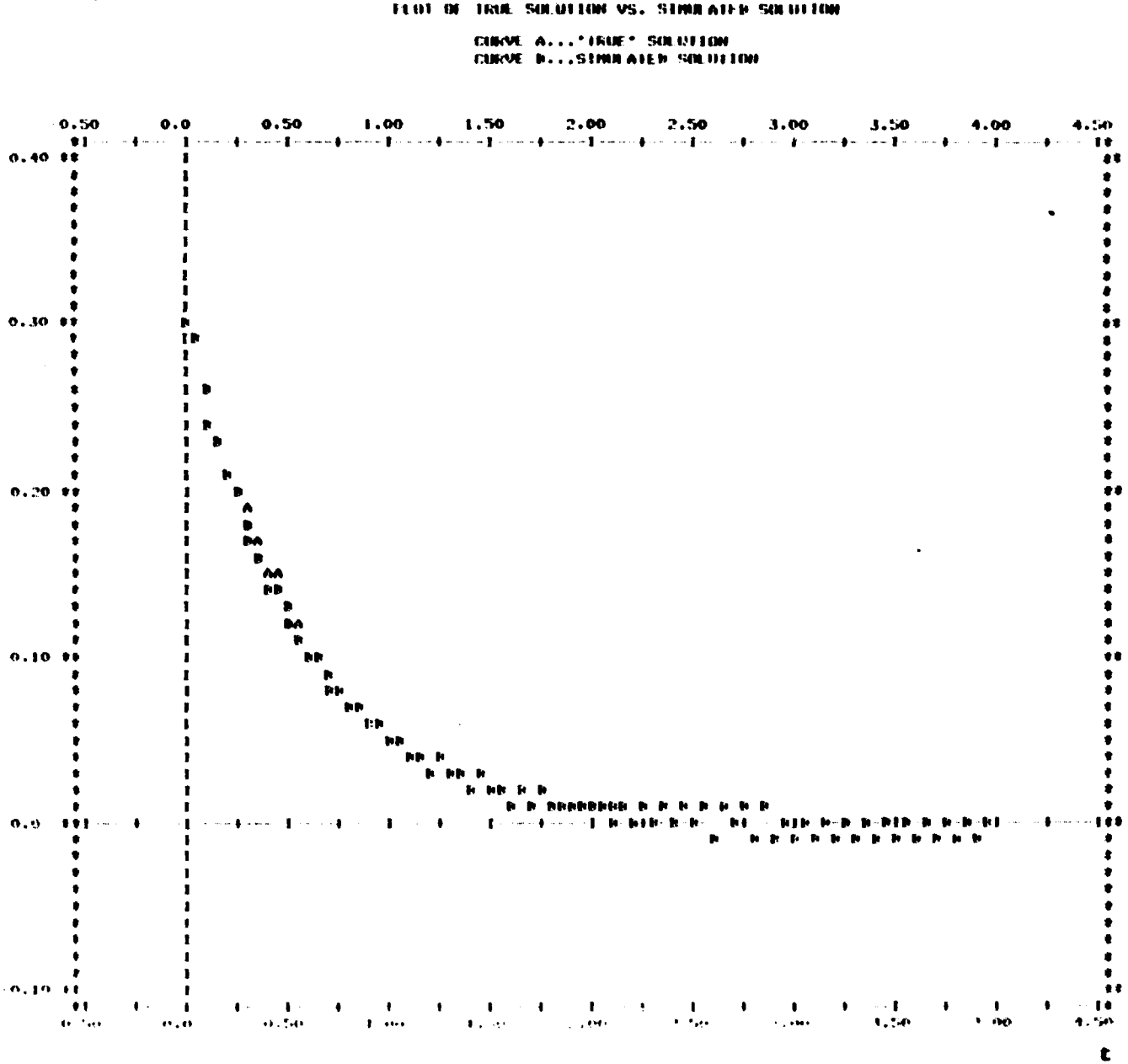



Figure 2.9

$$= u_1 u_2 \cosh(x_1 x_2^2) - e^{u_1} \sinh(2x_1) - 5 \sinh x_2;$$

$$\dot{x}_2 = f_2(x,u)$$

$$= e^{u_1 u_2} \sinh(x_1) - u_2^2 \cosh(x_1 x_2) + \sinh(x_2).$$

Initial conditions used in the identification will be

$$x_1(0) = 0.05,$$

$$x_2(0) = 0.075.$$

The forcing functions used in the initial integration are given by

$$u_1(t) = 0.2 \sin(2\pi f_1 t),$$

$$u_2(t) = 0.3 \sin(2\pi f_2 t).$$

where $f_1 = 5$ hertz and $f_2 = 8$ hertz. Note that the two conditions demanded of the example are satisfied;

$$f(0, 0) = 0 ,$$

and

$$L_{10} = \left[\begin{matrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{matrix}\right]\Bigg|_{\substack{x = 0 \\ u = 0}}$$

$$= \begin{bmatrix} -2 & -5 \\ 1 & 1 \end{bmatrix}$$

has eigenvalues in the left half complex plane, thus satisfying the stability requirement.

To begin the identification scheme, the original system is first in-

tegrated, with a stepsize of 0.005 again, given the above-mentioned operating point and input specifications. From this block of data, 40 samples are taken at 0.025 intervals, and third degree tensor terms are retained. In this case, the $\tilde{L}_{10}$ partition returned by the least-squares identification,

$$\tilde{L}_{10} = \begin{bmatrix} -1.526 & -8.210 \\ 1.005 & 1.262 \end{bmatrix}$$

is similar to the analytical result derived above, and has eigenvalues with negative real parts. When the identified system is simulated (again, it consists of two equations with 34 terms each in the sum), the results are very good for the region up to one second, as depicted in Figure 2.10 for $x_1$ and Figure 2.11 for $x_2$. This behavior is expected since the true solution was sampled on exactly this interval. But an extended look at these solutions shows an unacceptable tracking error beyond the transient region. In fact, for this system (as compared to the first example) the transient behavior takes longer to settle out due to the relative size of the eigenvalues of $\tilde{L}_{10}$.

The next step in the overall identification, then, is to increase the interval over which samples are taken so as to encompass complete information of the transient region. This can be done in one of two ways: either the sampling rate may be decreased (sampling period increased); or, the number of samples taken may be increased. Accuracy might suffer with the former, while the latter choice could increase computational complexity.

When the number of sample points is increased to 80 with the same sampling period of 0.025, the region of the true solution to be observed is from $t = 0$ to $t = 2$ seconds. Thus a larger portion, if not all, of the transient region would be sampled. Using the same initial condi-

tions and the same specifications for the control inputs, the first partition of the identified system

$$\hat{L}_{10} = \begin{bmatrix} -2.023 & -5.093 \\ 1.009 & 0.941 \end{bmatrix}$$

shows much better agreement with the analytical expression for $L_{10}$. Furthermore, the tracking is greatly improved, as illustrated in Figures 2.12-2.13 for $x_1$ and $x_2$, respectively. Note that in each solution curve B overlays curve A exactly for the region up to two seconds. But there is a slight tracking error beyond this point which remains approximately constant in time after three seconds. This constant steady-state error would be acceptable in many applications, although more accuracy is possible.

A sensitivity analysis on this system as identified for $x_1(0) = 0.05$ and $x_2(0) = 0.075$ shows good results when the initial conditions are varied in magnitude, that is,

$$x_1(0) = 0.10,$$
$$x_2(0) = 0.03.$$

The results using this initial condition (with the model system identified for the original initial condition) to simulate a corresponding true solution are given in Figures 2.14-2.17. For the variable $x_1$, Figure 2.14 shows a good overlay of curve B onto curve A, with a small error arising around the peak in the solution. Figure 2.16 shows a plot over a wider range of time, and reveals that the simulated solution (for $x_1$ again) actually improves past four seconds. Solution $x_2$, represented in Figures 2.15 and 2.17, does not exhibit quite as good results, but still maintains an acceptable curve fit with constant error.

The accuracy of these simulations can be improved upon even more by

sampling the original system over a yet wider region. So consider a sampling scheme which again uses 80 sample points, but which now takes samples at a rate of 25 per second as opposed to 40 per second. This corresponds to a sampling period of 0.040 seconds so that for 80 samples the region from $t = 0.0$ to $t = 3.2$ seconds is sampled. Figures 2.18 and 2.19 show solutions $x_1$ and $x_2$ out to four seconds for the model system using those initial conditions as used in the identification (again taken to be $x_1(0) = 0.05$ and $x_2(0) = 0.075$). The tracking is very good for this system now, as in each solution curve B overlays curve A throughout. Now a sensitivity analysis is performed by simulating with the system model using $x_1(0) = 0.10$ and $x_2(0) = 0.03$. A comparison of Figure 2.16 to Figure 2.20 for solution $x_1$ reveals that for the latter plot, which corresponds to the model system identified from samples taken to 3.2 seconds, the match of the simulated solution to the true solution is much better. This is particularly noticeable in Figure 2.21 for $x_2$, upon comparison to Figure 2.17 for the previous identification. It would seem that even more accuracy could be achieved, although this last pair of plots indicates that a good system model has been achieved without suffering any additional ill effects in computational complexity. In fact, it is evident that the decrease in sampling rate in the last identification caused no noticeable error; to the contrary, it led to more accurate results.

These two second order examples presented in this section were discussed in detail to illustrate the flexibility of the method. The first, an exact vector differential equation, was chosen to give some meaning to the elements of the individual $\tilde{L}_{jk}$ operators and to illustrate that the degree of the approximation is in general problem intensive. The second

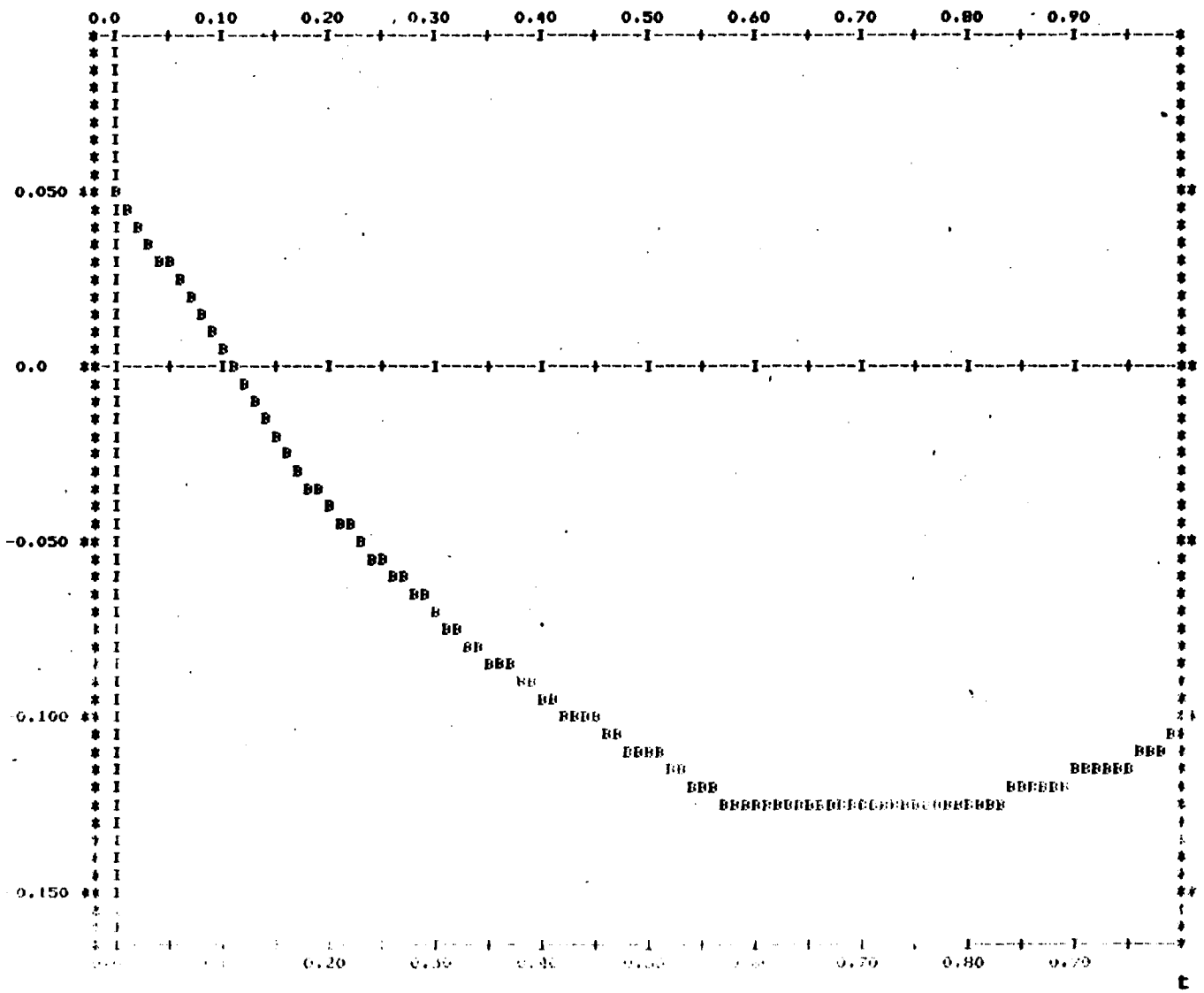



Figure 2.10

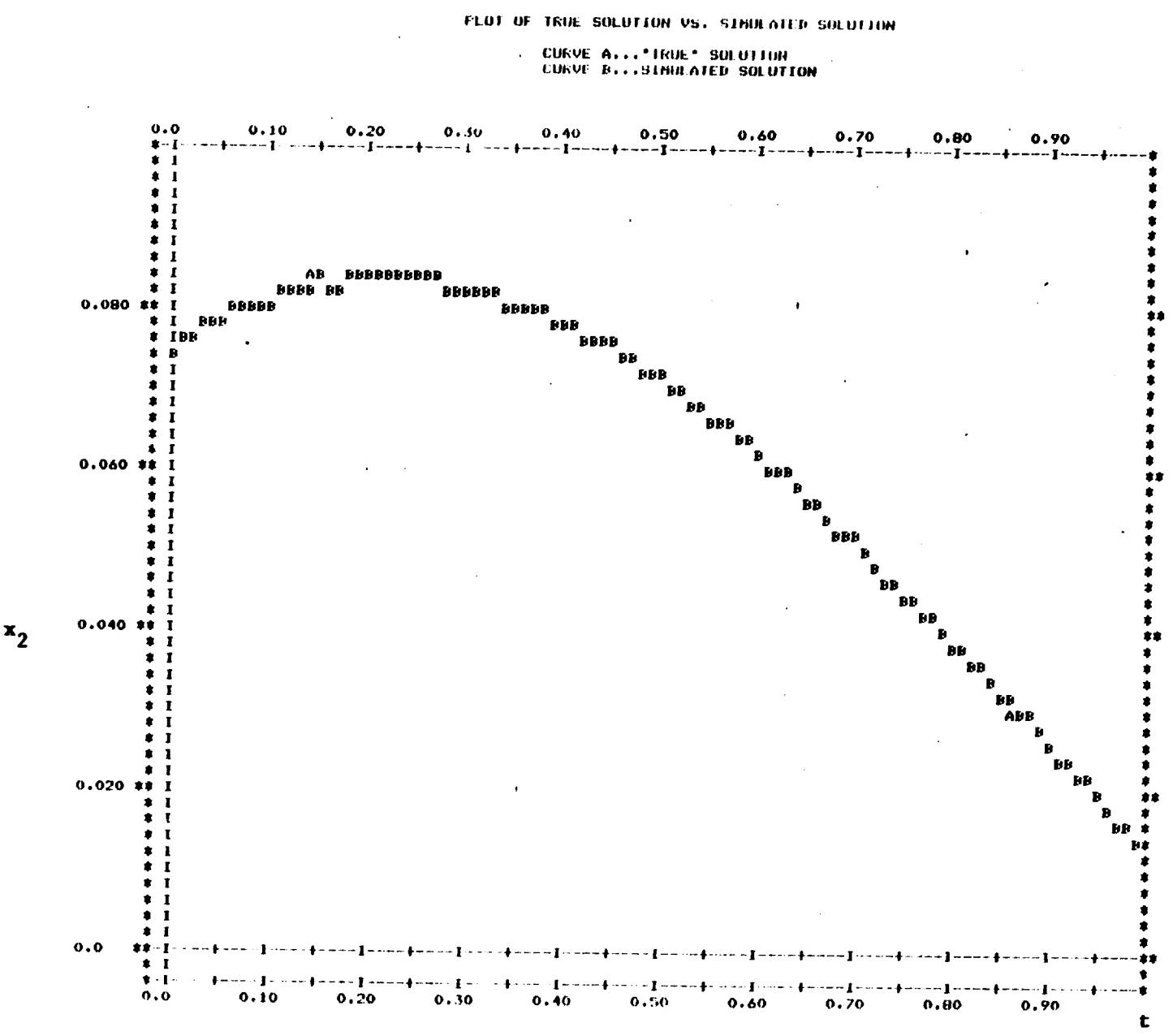

PLOT OF TRUE SOLUTION VS. SIMULATED SOLUTION
CURVE A...'TRUE' SOLUTION
CURVE B...SIMULATED SOLUTION
0.0
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
0.080
0.060
0.040
0.020
0.0
$x_2$
t

Figure 2.11

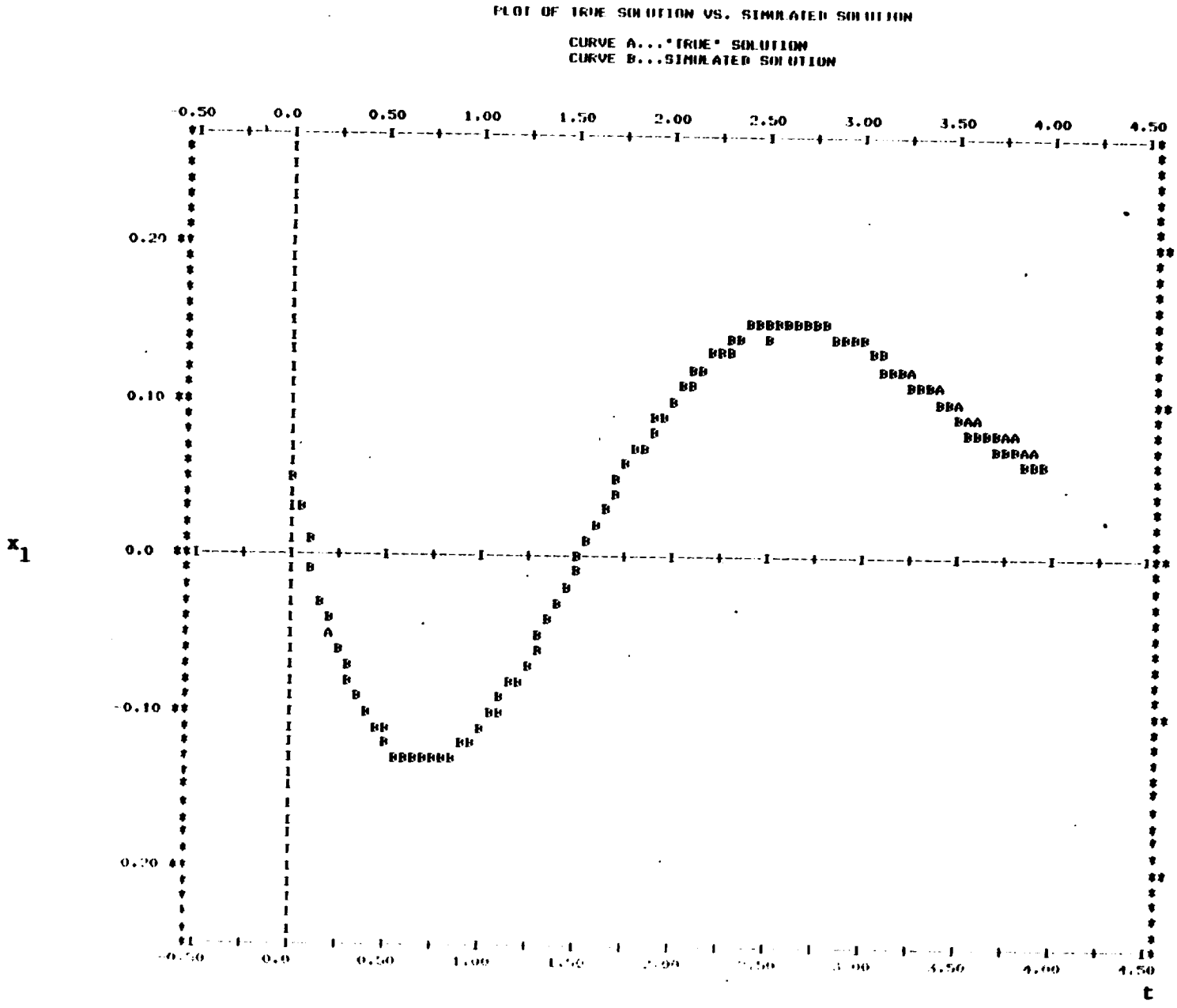

PLOT OF TRUE SOLUTION VS. SIMULATED SOLUTION
CURVE A...'TRUE' SOLUTION
CURVE B...SIMULATED SOLUTION
-0.50
0.0
0.50
1.00
1.50
2.00
2.50
3.00
3.50
4.00
4.50
0.20
0.10
0.0
-0.10
-0.20
x1
t

Figure 2.12

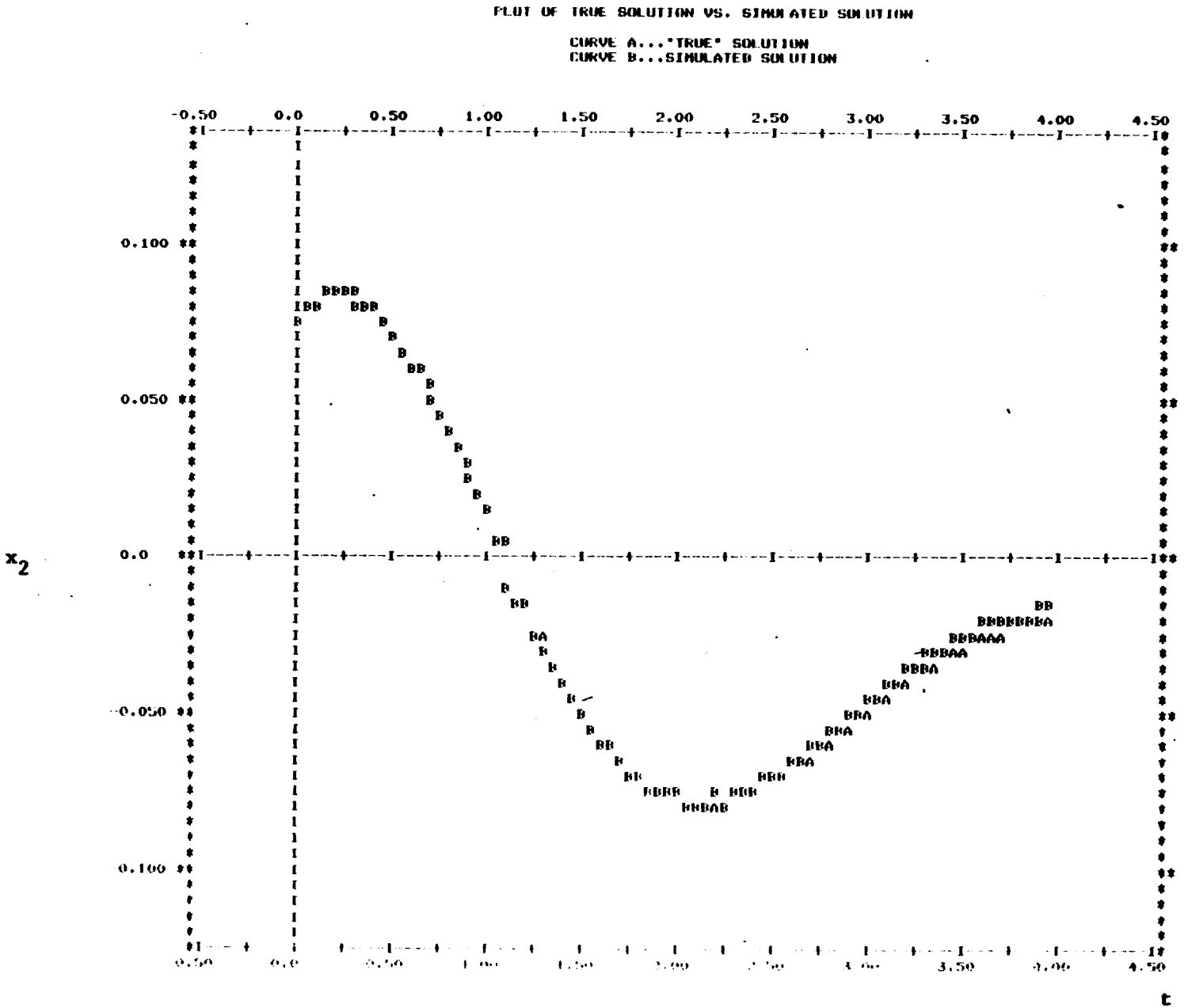

PLOT OF TRUE SOLUTION VS. SIMULATED SOLUTION
CURVE A...'TRUE' SOLUTION
CURVE B...SIMULATED SOLUTION
-0.50
0.0
0.50
1.00
1.50
2.00
2.50
3.00
3.50
4.00
4.50
0.100
0.050
0.0
-0.050
-0.100
$x_2$
t

Figure 2.13

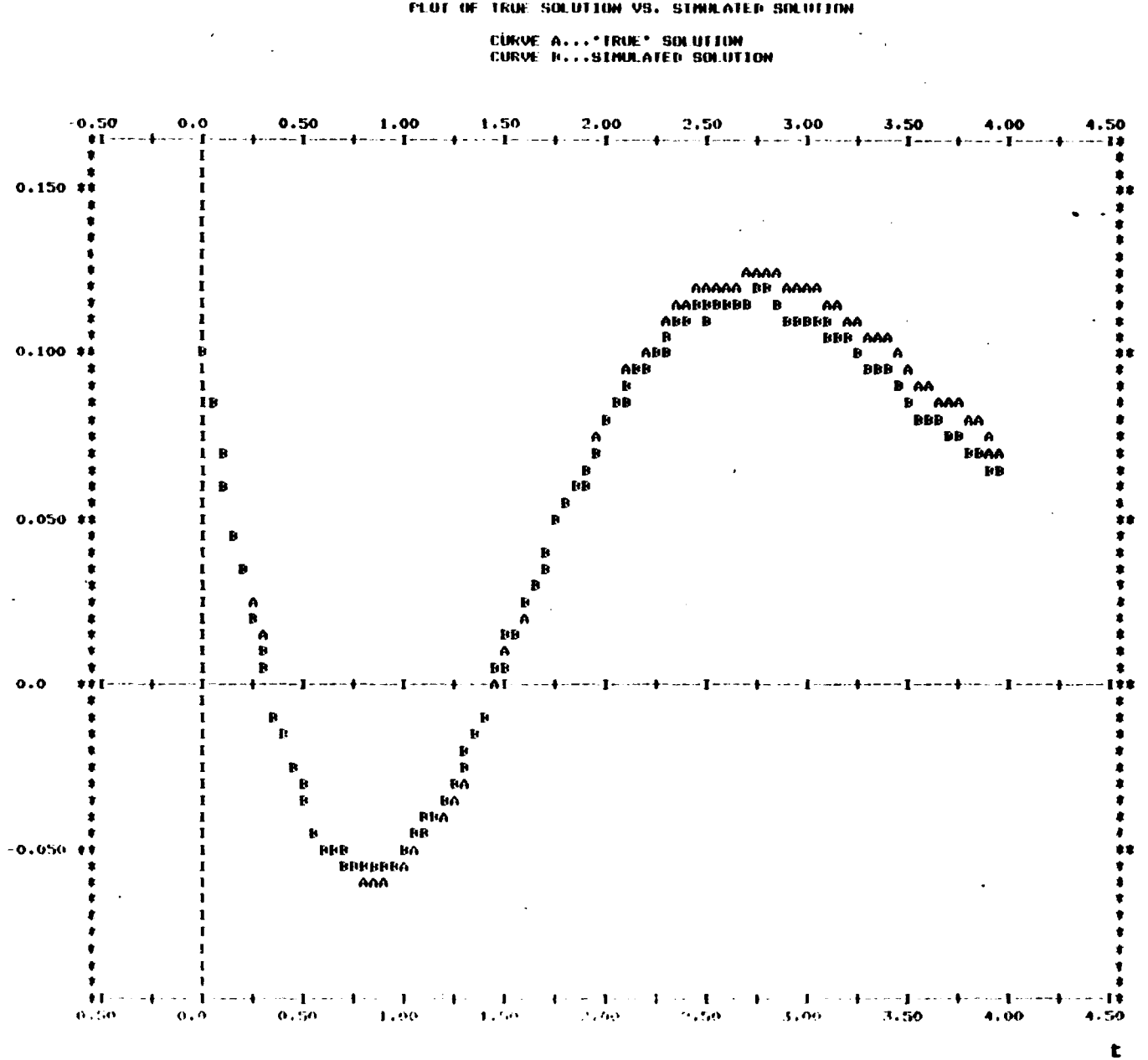

PLOT OF TRUE SOLUTION VS. SIMULATED SOLUTION
CURVE A...'TRUE' SOLUTION
CURVE B...SIMULATED SOLUTION
-0.50
0.0
0.50
1.00
1.50
2.00
2.50
3.00
3.50
4.00
4.50
0.150
0.100
0.050
0.0
-0.050
$x_1$
t

Figure 2.14

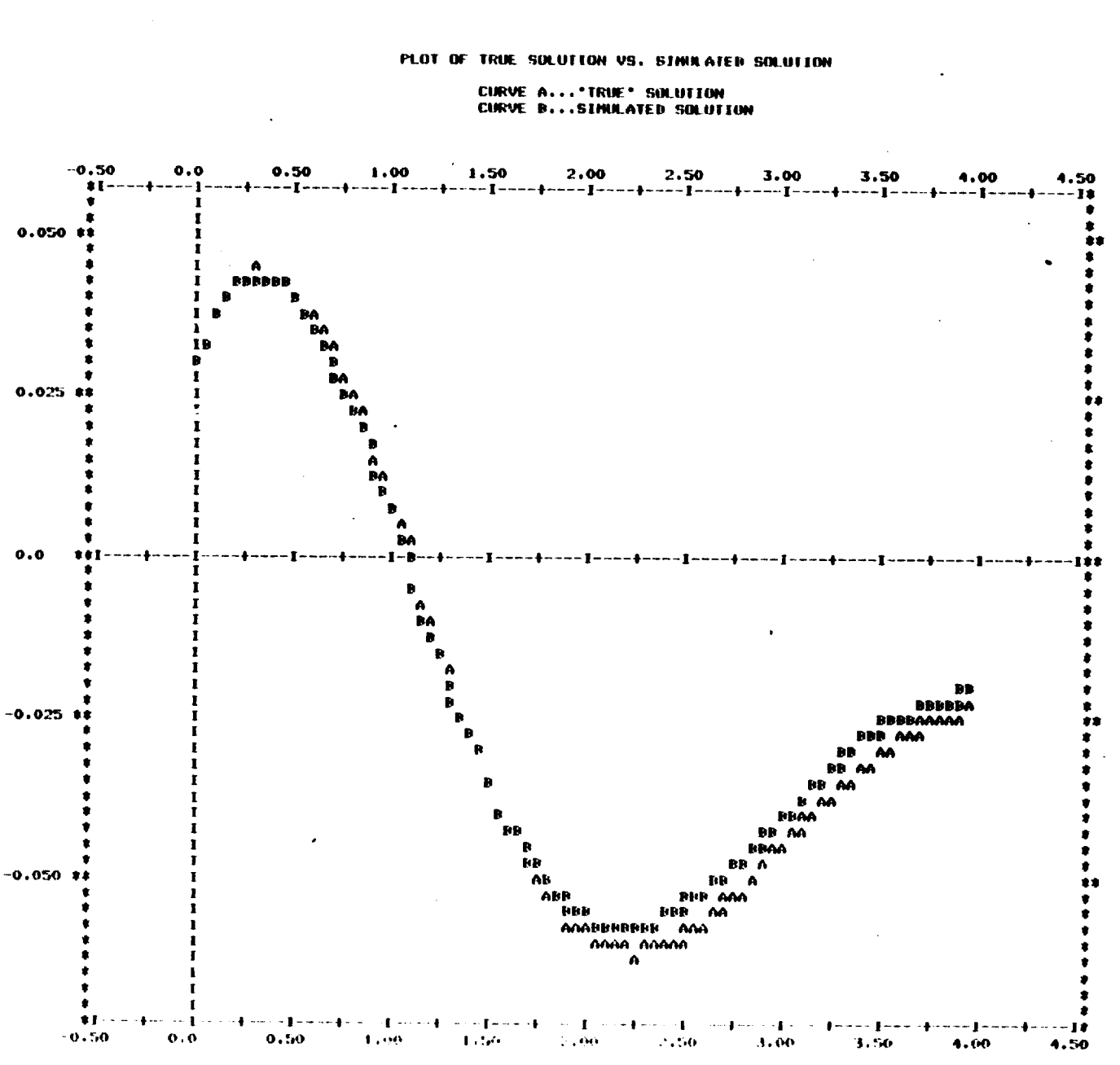



Figure 2.15

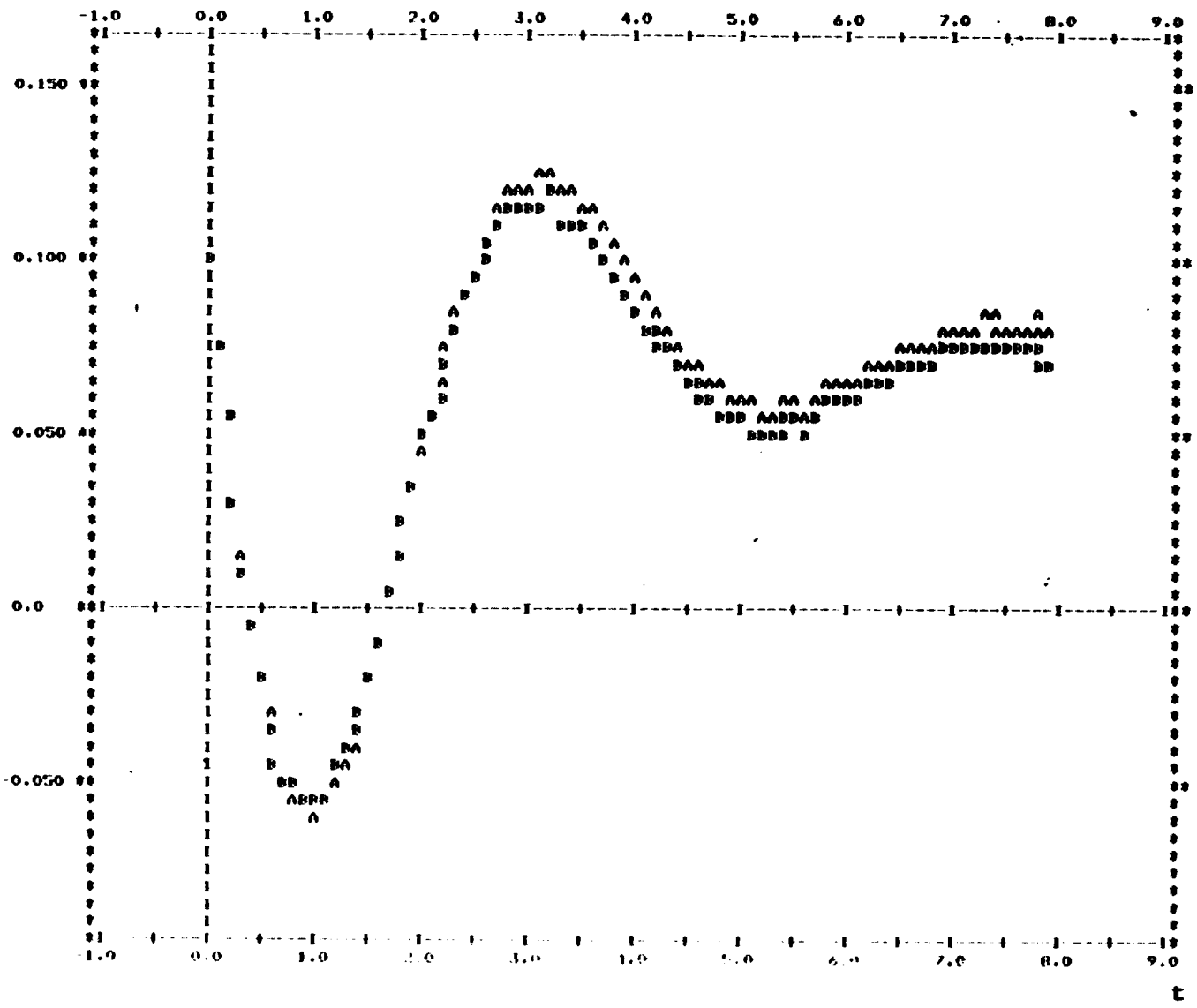

PLOT OF TRUE SOLUTION VS. SIMULATED SOLUTION
CURVE A...'TRUE' SOLUTION
CURVE B...SIMULATED SOLUTION
-1.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
0.150
0.100
0.050
0.0
-0.050
$x_1$
t

Figure 2.16

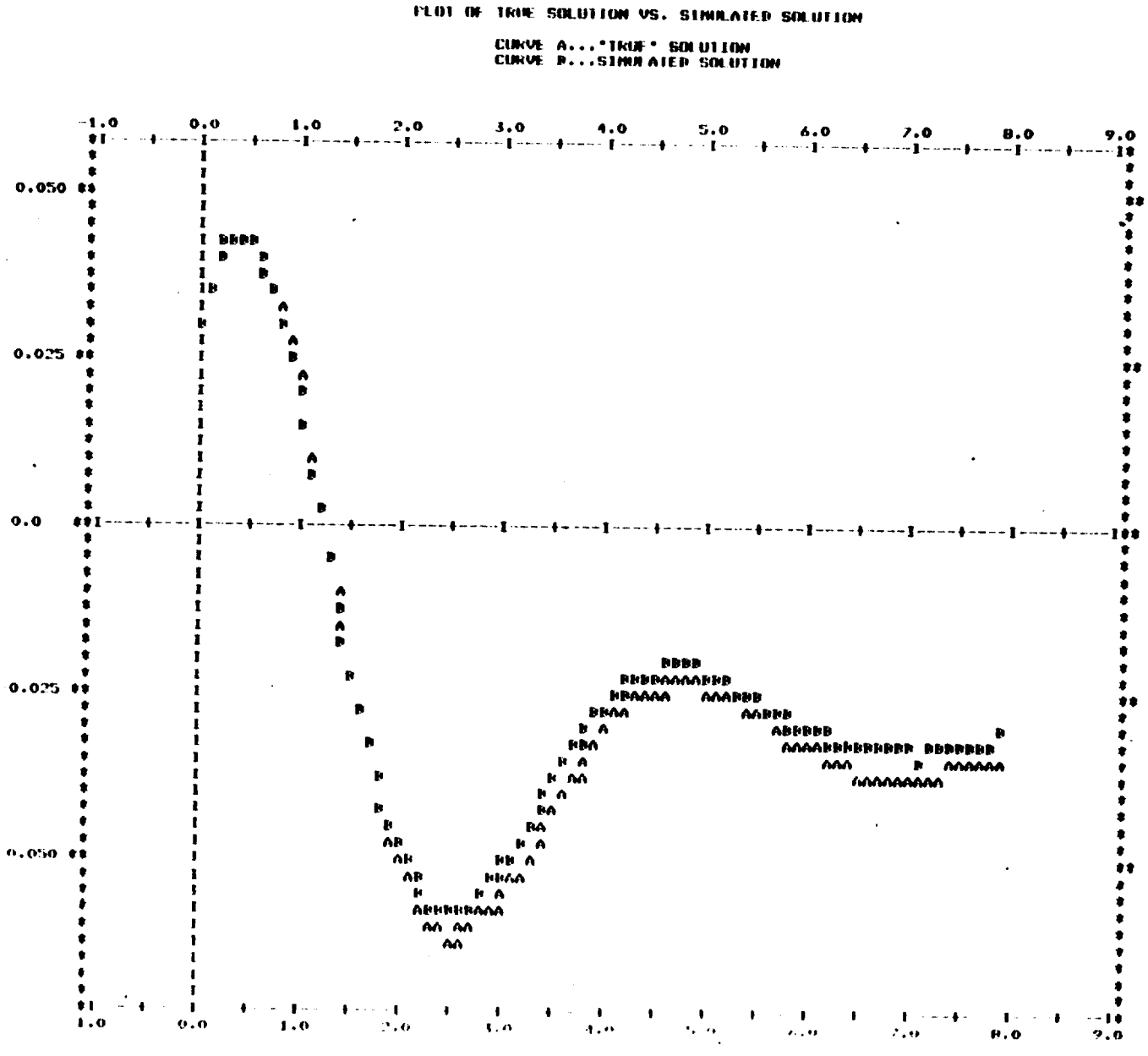

PLOT OF TRUE SOLUTION VS. SIMULATED SOLUTION
CURVE A...'TRUE' SOLUTION
CURVE B...SIMULATED SOLUTION
-1.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
0.050
0.025
0.0
0.025
0.050
$x_2$
t

Figure 2.17

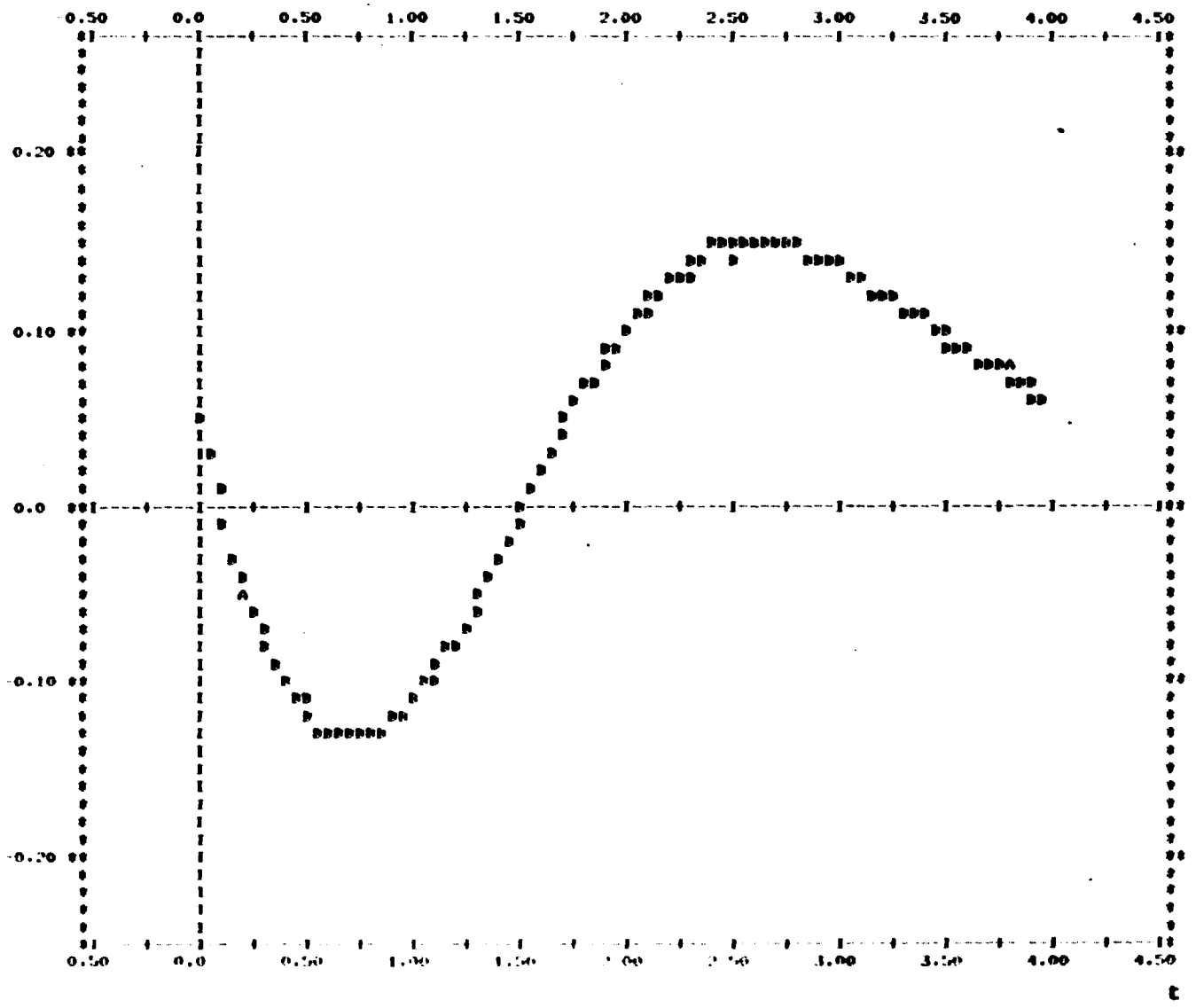



Figure 2.18

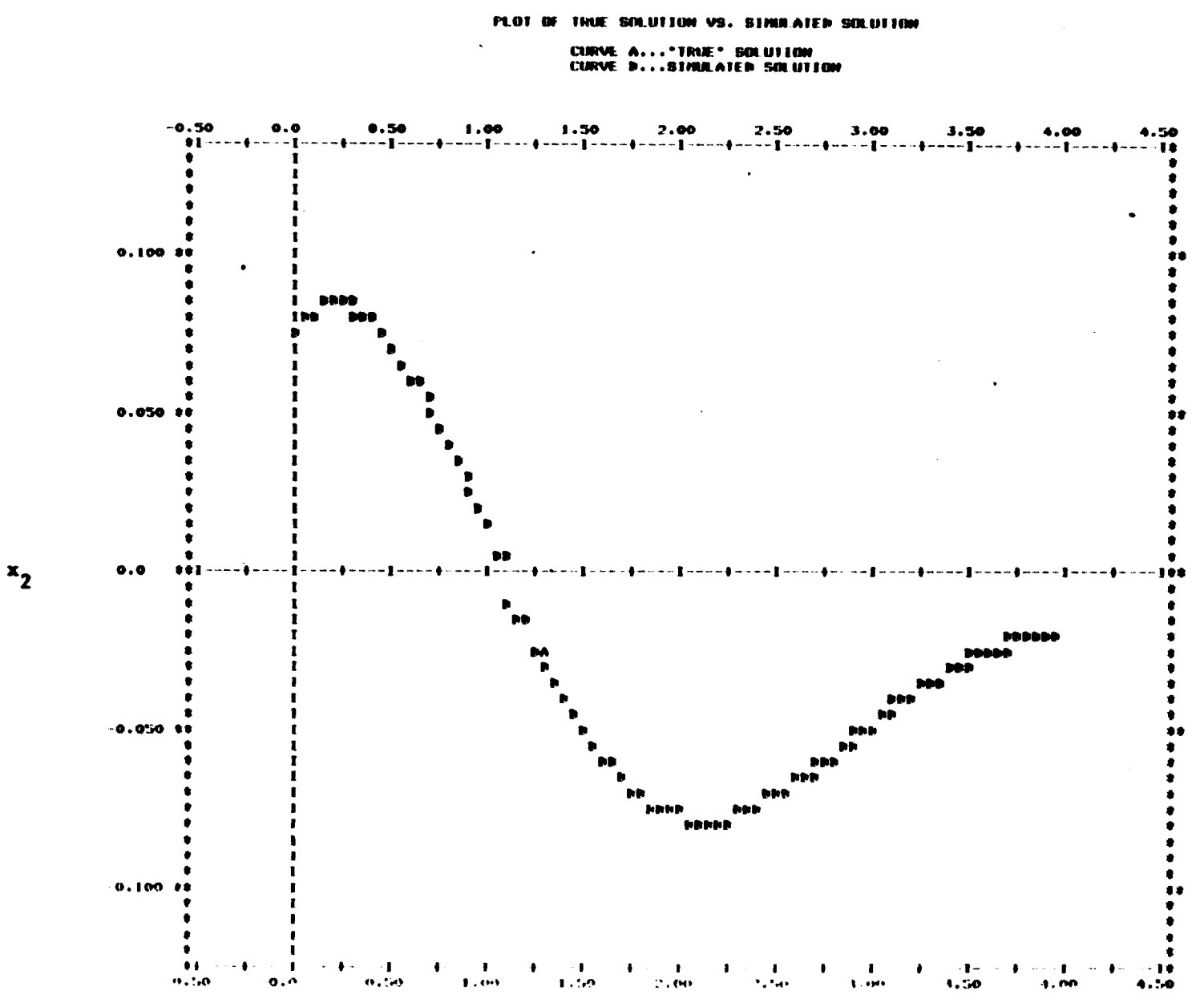

PLOT OF TRUE SOLUTION VS. SIMULATED SOLUTION
CURVE A...'TRUE' SOLUTION
CURVE B...SIMULATED SOLUTION
-0.50
0.0
0.50
1.00
1.50
2.00
2.50
3.00
3.50
4.00
4.50
0.100
0.050
0.0
-0.050
-0.100
S1
$x_2$
t

Figure 2.19

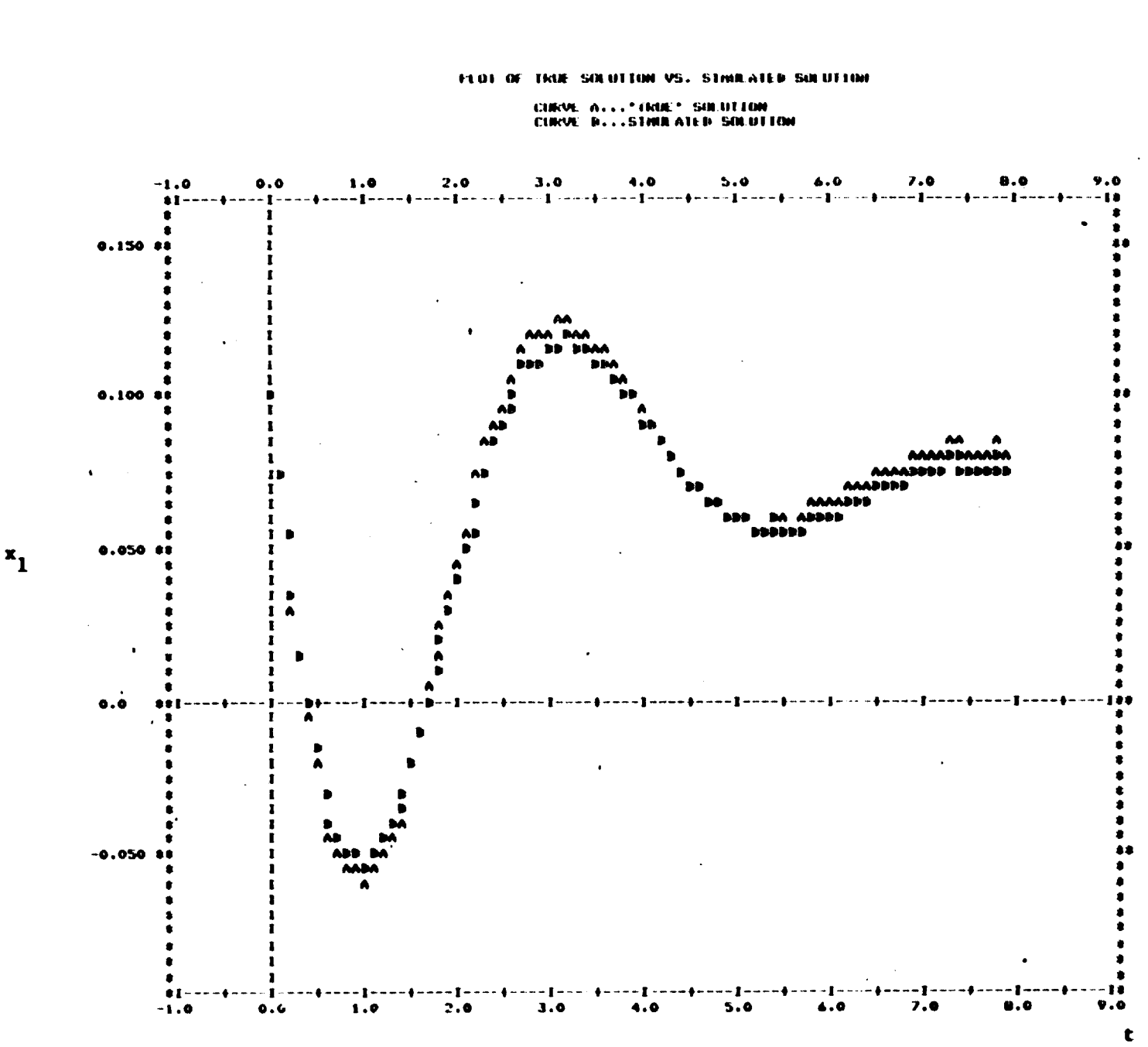
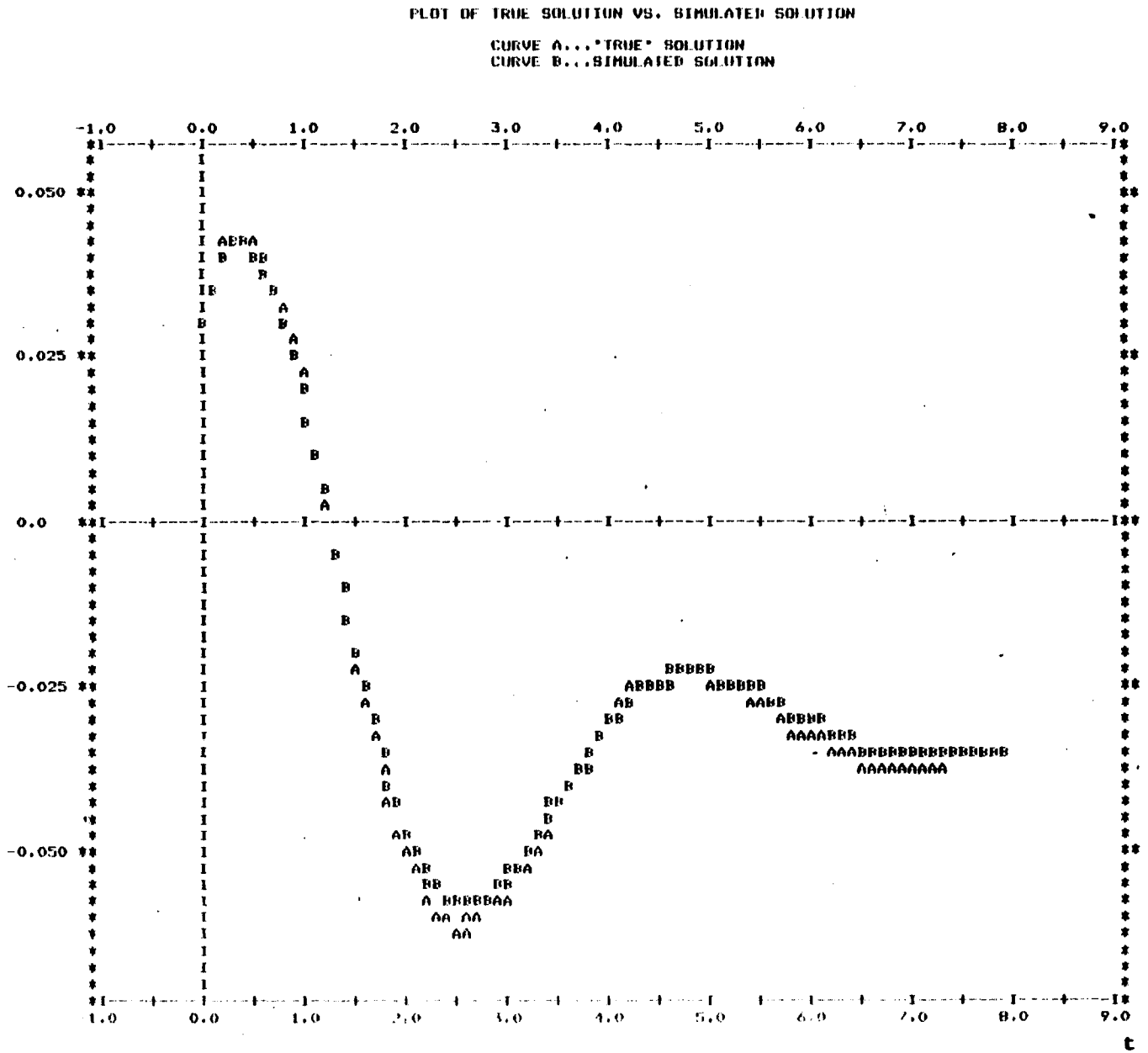

PLOT OF TRUE SOLUTION VS. SIMULATED SOLUTION
CURVE A...'TRUE' SOLUTION
CURVE B...SIMULATED SOLUTION
-1.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
0.150
0.100
0.050
0.0
-0.050
x1
t

Figure 2.20




$x_2$

Figure 2.21

example showed that even a very unattractive nonlinear, nonhomogeneous system can be adequately handled by this technique.

### 2.1.4 Third Order Example

This section treats an example given by a system of three nonhomogeneous differential equations. The state vector,

$$x = (x_1, x_2, x_3)',$$

consists of three elements, as does

$$u(t) = (u_1(t), u_2(t), u_3(t))',$$

the input vector of forcing functions. Obviously, the size of the problem will be increased compared to that of the second order systems of two states and two inputs. Whereas for $n = m = 2$ in the previous examples the length of the tensor term vector $x_L$ was 34, now the length of $x_L$ will be $p = 83$ for an approximation keeping up to third degree terms. Again, this number is calculated according to

$$p = \sum_{i=1}^{9} \left[\binom{n+q-1}{q} \cdot \binom{m+r-1}{r}\right]_i$$

for the $i$ partitions. Expanding this for $n = 3$ (three states) and $m = 3$ (three inputs), the result is

$$p = 3 + 3 + 6 + 9 + 6 + 10 + 18 + 18 + 10$$
$$= 83.$$

Thus, the matrix equation to be constructed takes the following form with matrix dimensions as indicated:

$$\underset{3xh}{\dot{X}} = \underset{3x83}{[\tilde{L}_{10}\ \tilde{L}_{01}\ \tilde{L}_{20}\ \tilde{L}_{11}\ \tilde{L}_{02}\ \tilde{L}_{30}\ \tilde{L}_{21}\ \tilde{L}_{12}\ \tilde{L}_{03}]}\ \underset{83xh}{X_L}$$

Here, $h$ is the number of sample points used in the loading scheme for

estimation of derivatives and construction of $X_L$.

The third order system of this example is of the same nature as the previous example in that exponential and hyperbolic terms comprise the non-linear system. Consider the following equations,

$$f_1(x,u) = \dot{x}_1$$
$$= u_1 e^{u_2} \cosh(x_1 x_3) - e^{u_3} \sinh(3x_1);$$

$$f_2(x,u) = \dot{x}_2$$
$$= u_1^2 \cosh(x_1^2) - 2 \sinh(x_2) + u_1 u_3 \cosh(x_3);$$

$$f_3(x,u) = \dot{x}_3$$
$$= u_1 \sinh(x_1 x_2) - \sinh(x_3).$$

To identify the model system, the initial conditions chosen are given by

$$x_1(0) = 0.05,$$
$$x_2(0) = -0.05,$$
$$x_3(0) = 0.08.$$

Choices for forcing function inputs are sinusoids with the specifications

$$u_1(t) = \sin(2\pi f_1 t),$$
$$u_2(t) = (0.5) \sin(2\pi f_2 t),$$
$$u_3(t) = (0.5) \sin(2\pi f_3 t),$$

where the frequencies for each input are given by

$$f_1 = 1 ,$$
$$f_2 = 5 ,$$
$$f_3 = 10,$$

in hertz.

This system is constructed such that the following conditions are easily verified:

$$f(0, 0) = 0 ;$$

and for stability

$$L_{10} = \left[\frac{\partial f_i}{\partial x_j}\right]\Bigg|_{\substack{x = 0 \\ u = 0}}$$

$$= \begin{bmatrix} -3 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -1 \end{bmatrix} .$$

With the problem thus formulated the technique is applied. Using an integration stepsize of 0.005, the system is integrated and the true solution stored. This data is sampled at 40 time points evenly spaced at intervals of 0.025 for loading of the tensor term matrix and for derivative estimation. Thus, the sampling takes place over the first one second interval of the solution. The first partition returned by routine SIMEQUAT in the least-squares minimization is given by

$$\hat{L}_{10} = \begin{bmatrix} -3.072 & -0.112 & 0.440 \\ 0.009 & -1.948 & 0.003 \\ -0.001 & 0.005 & -0.936 \end{bmatrix}$$

which has eigenvalues with negative real parts. Note the similarity to the analytical expression for $L_{10}$ given above.

Simulations using this model system consist of an integration of the three equations which have 83 terms each in the sum. Over the interval in which the 40 samples were taken, the simulated solution matches the true solution very well when the initial conditions are those as used in the identification. These results are shown in Figures 2.22-2.24 for $x_1, x_2,$

and $x_3$, respectively. Figures 2.25-2.27 depict the same simulations over a wider range of time. The most error occurs at the peaks of the solution curves, especially noticeable in the variable $x_1$ (Figure 2.25).

Consider now a sensitivity analysis on this model system for the following changes in the initial state conditions:

$$x_1(0) = 0.10 ,$$
$$x_2(0) = -0.10 ,$$
$$x_3(0) = 0.16 .$$

Using the same forcing function as in the identification, the original system is integrated with these new initial conditions to form the true solution. Then the model system is employed with these initial conditions to simulate the true solution. These results are shown in Figures 2.28-2.30 over an eight second interval of time, well beyond the transient region of each solution. Very little error results between the simulated solution and true solution.

A conclusion of the final three plots is that for this particular example an ample number of sample points for system identification is 40; perhaps even fewer would produce equally acceptable results. In the final analysis the initial condition of each variable is double that which was used to identify the model system, indicating (by the accuracy of the tracking in the simulations) that an even greater variance in the initial conditions would produce acceptable results. Thus, this model system seems to have a larger feasible operating region than that of the examples treated in the previous section.

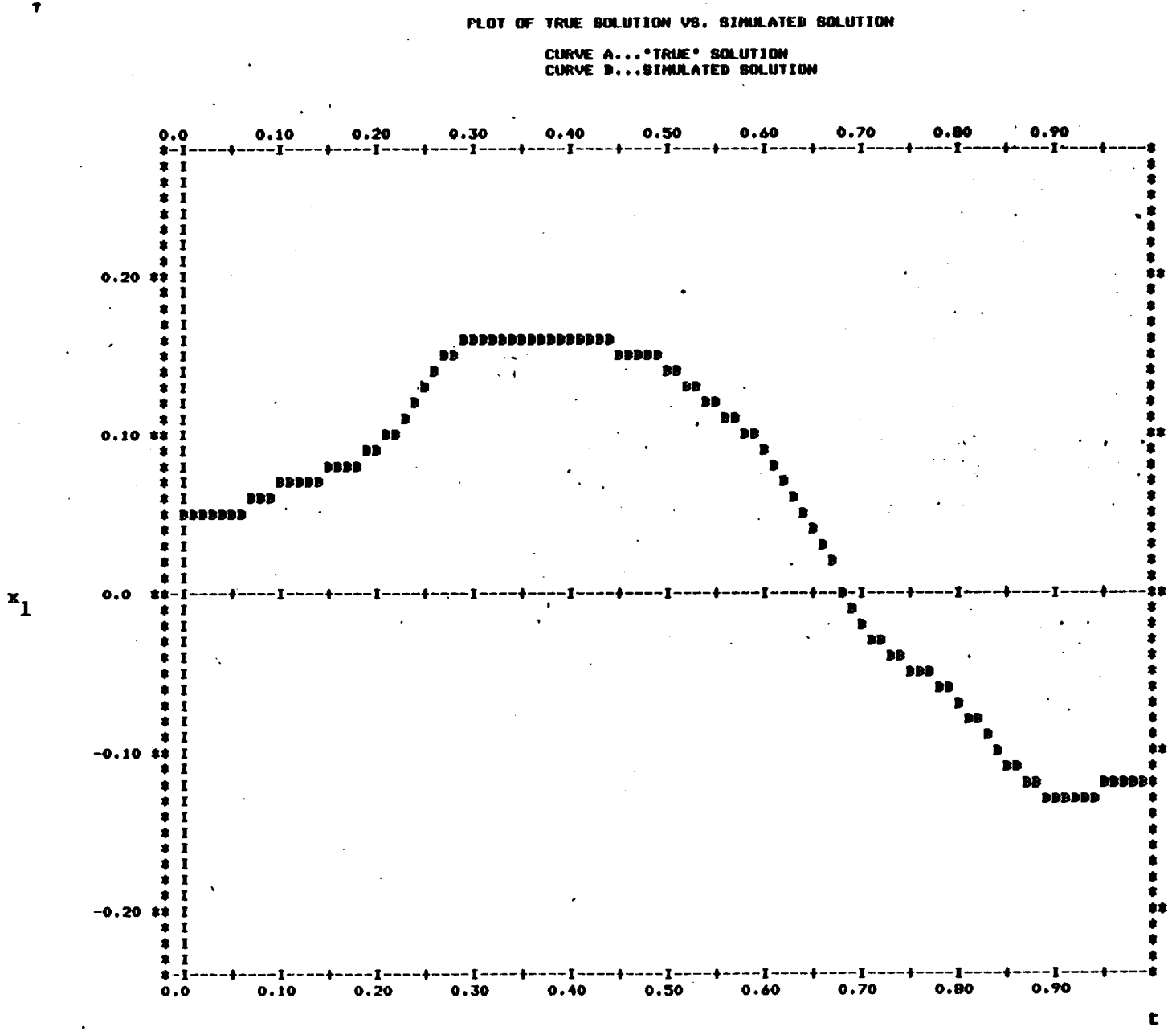

PLOT OF TRUE SOLUTION VS. SIMULATED SOLUTION
CURVE A...'TRUE' SOLUTION
CURVE B...SIMULATED SOLUTION
0.0
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
0.20
0.10
0.0
-0.10
-0.20
x1
t

Figure 2.22

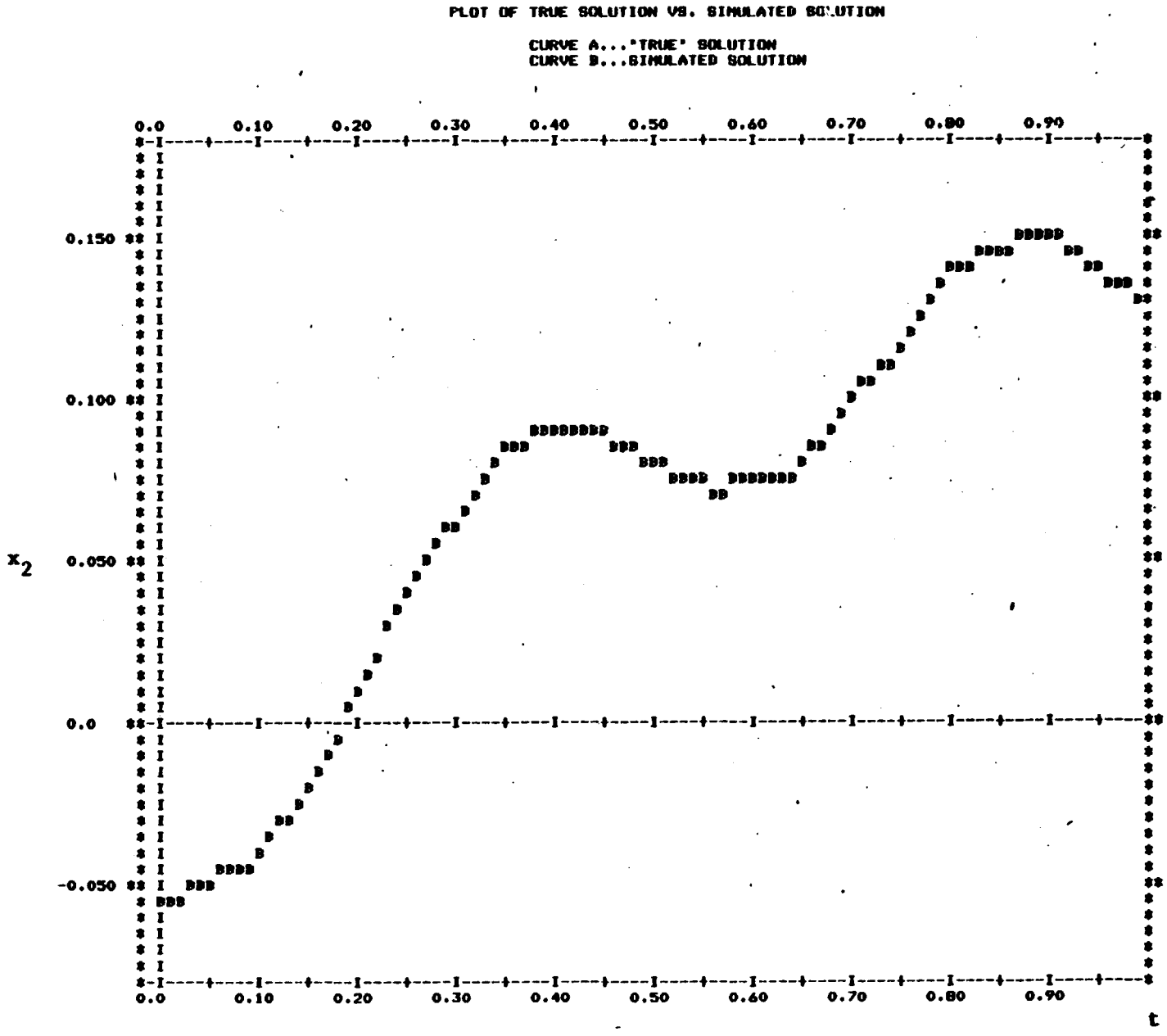

PLOT OF TRUE SOLUTION VS. SIMULATED SOLUTION
CURVE A...'TRUE' SOLUTION
CURVE B...SIMULATED SOLUTION
0.0
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
0.150
0.100
0.050
0.0
-0.050
$x_2$
t

Figure 2.23

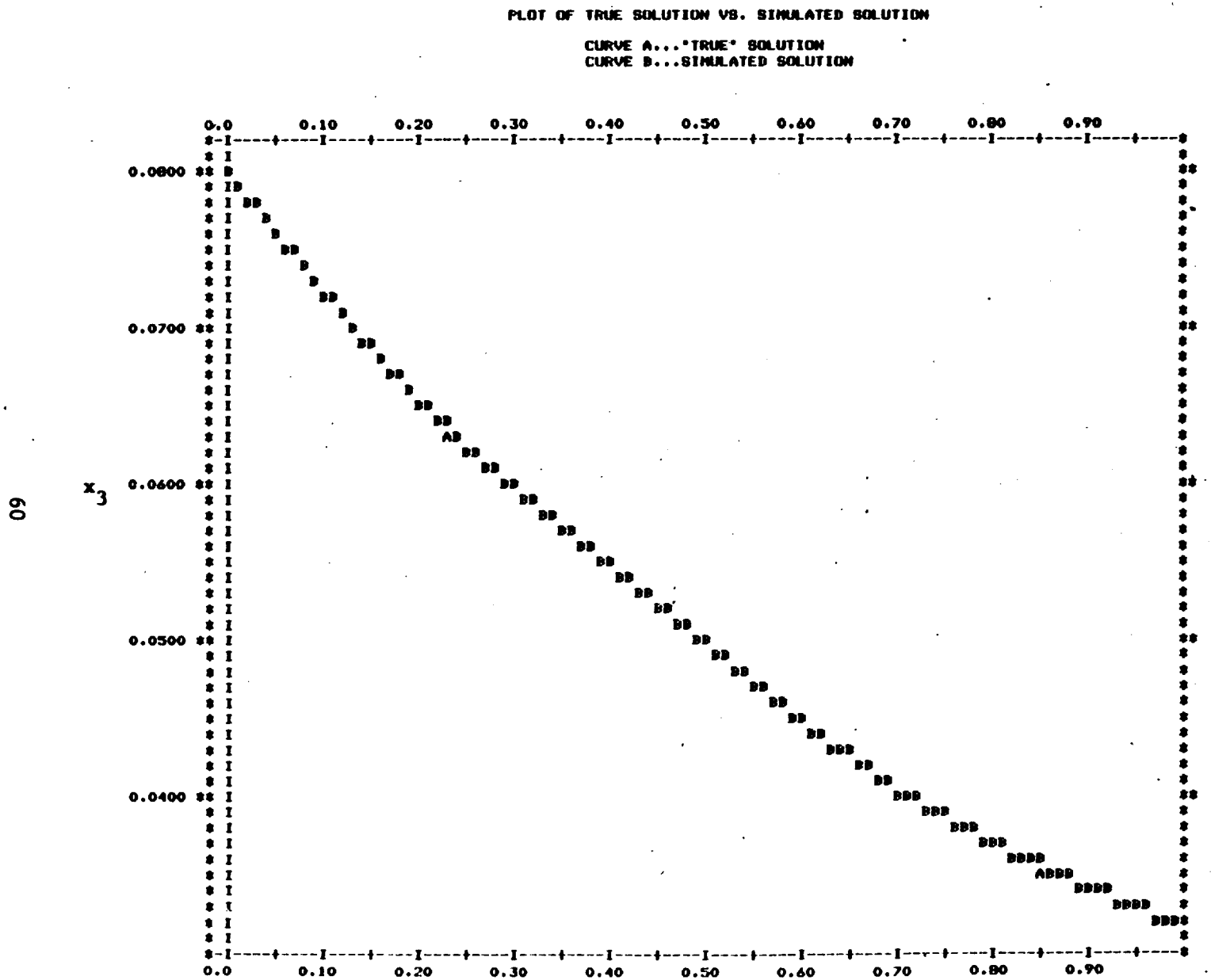

PLOT OF TRUE SOLUTION VS. SIMULATED SOLUTION
CURVE A...'TRUE' SOLUTION
CURVE B...SIMULATED SOLUTION
0.0
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
0.0800
0.0700
0.0600
0.0500
0.0400
$x_3$
t

Figure 2.24

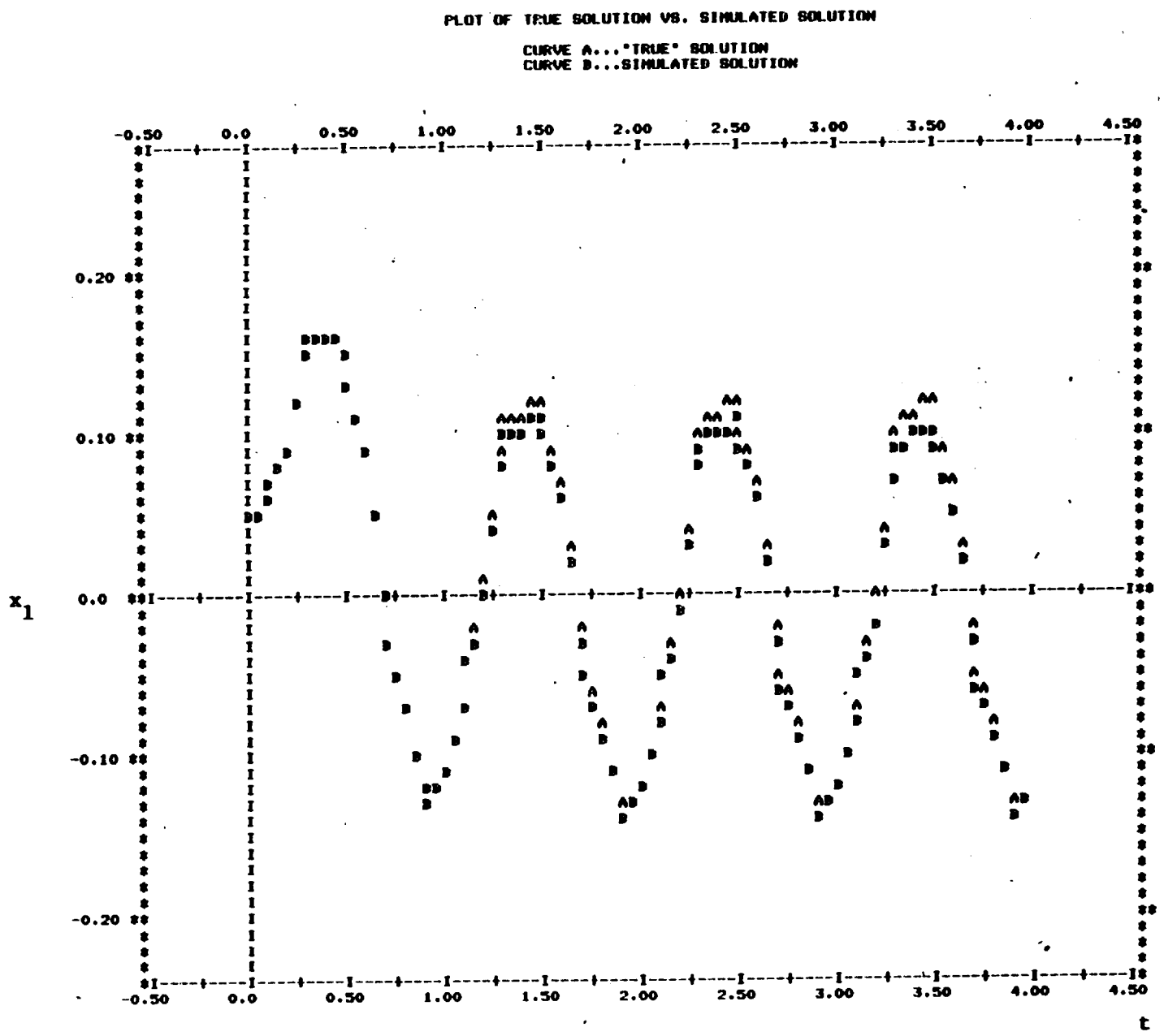

PLOT OF TRUE SOLUTION VS. SIMULATED SOLUTION
CURVE A..."TRUE" SOLUTION
CURVE B...SIMULATED SOLUTION
-0.50
0.0
0.50
1.00
1.50
2.00
2.50
3.00
3.50
4.00
4.50
0.20
0.10
0.0
-0.10
-0.20
$x_1$
t

Figure 2.25

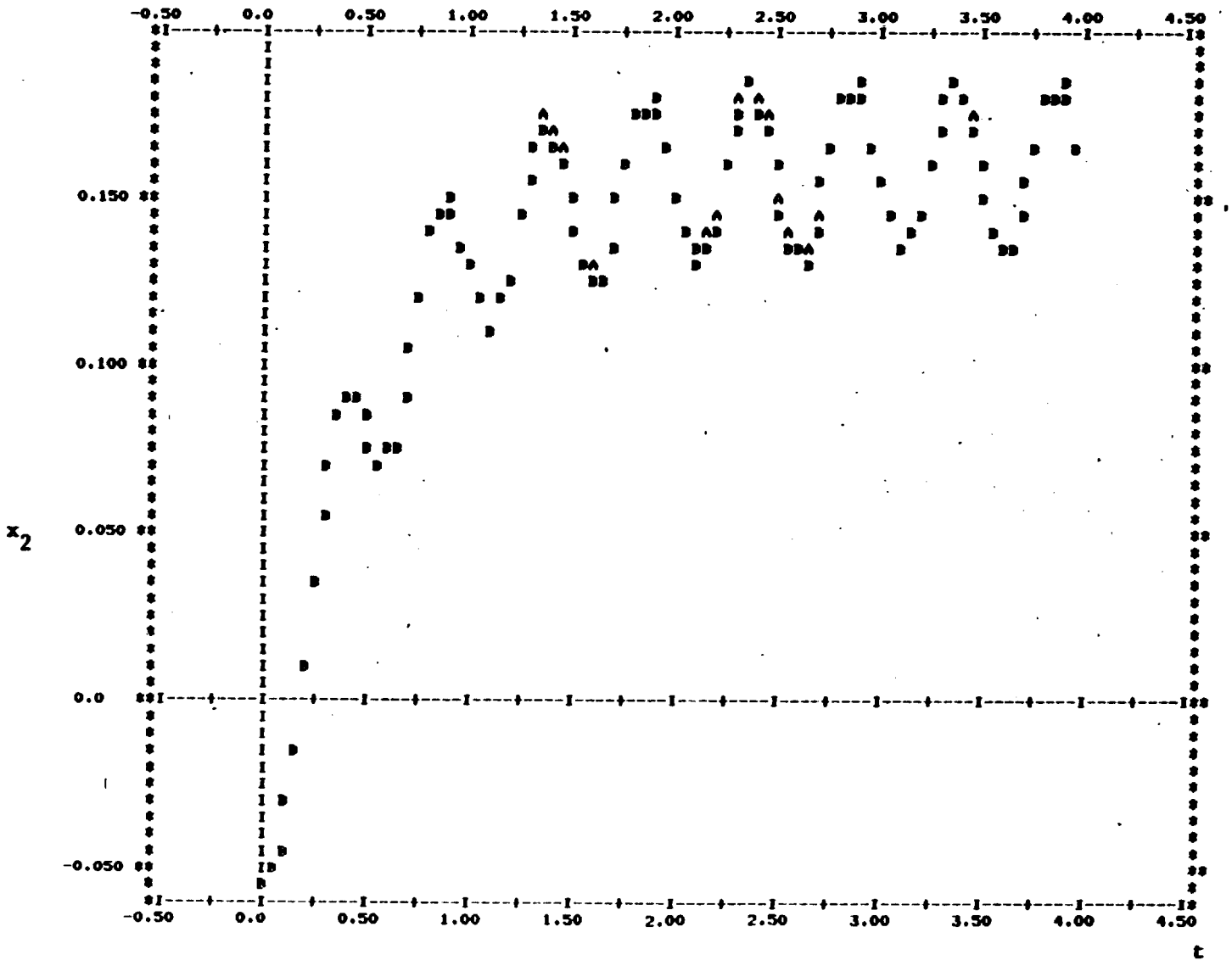

PLOT OF TRUE SOLUTION VS. SIMULATED SOLUTION
CURVE A...'TRUE' SOLUTION
CURVE B...SIMULATED SOLUTION
-0.50
0.0
0.50
1.00
1.50
2.00
2.50
3.00
3.50
4.00
4.50
0.150
0.100
0.050
0.0
-0.050
$x_2$
t

Figure 2.26

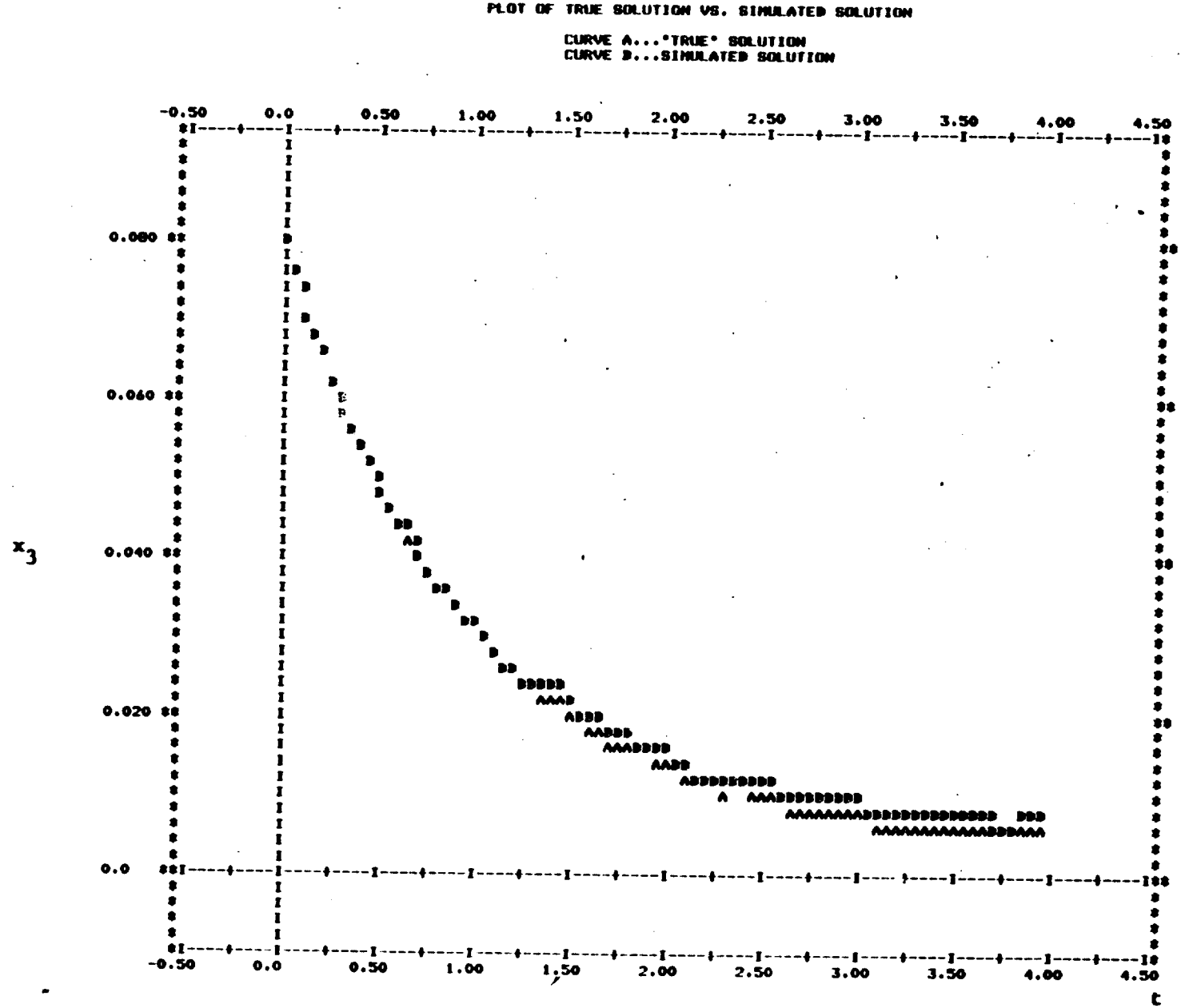
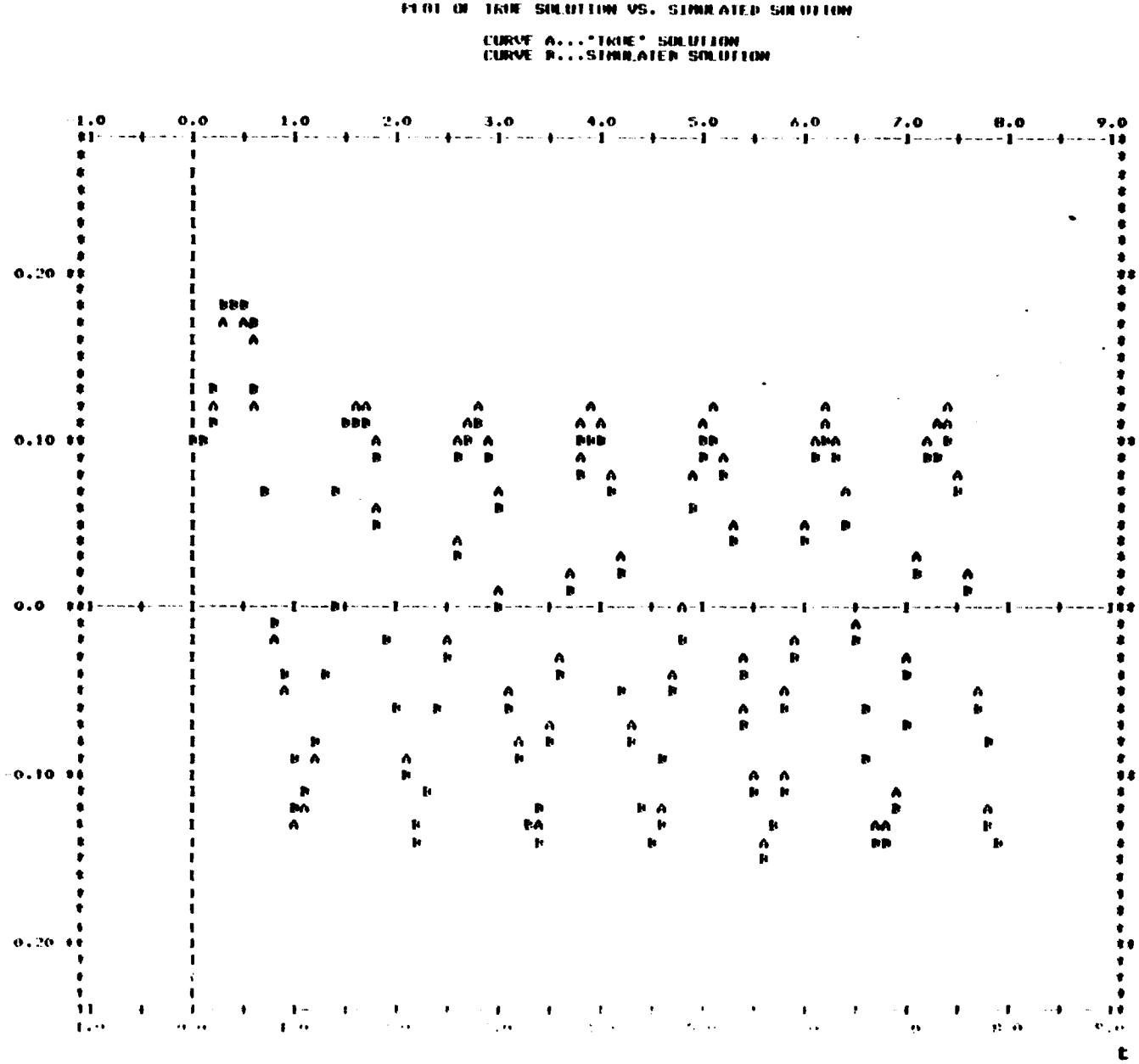
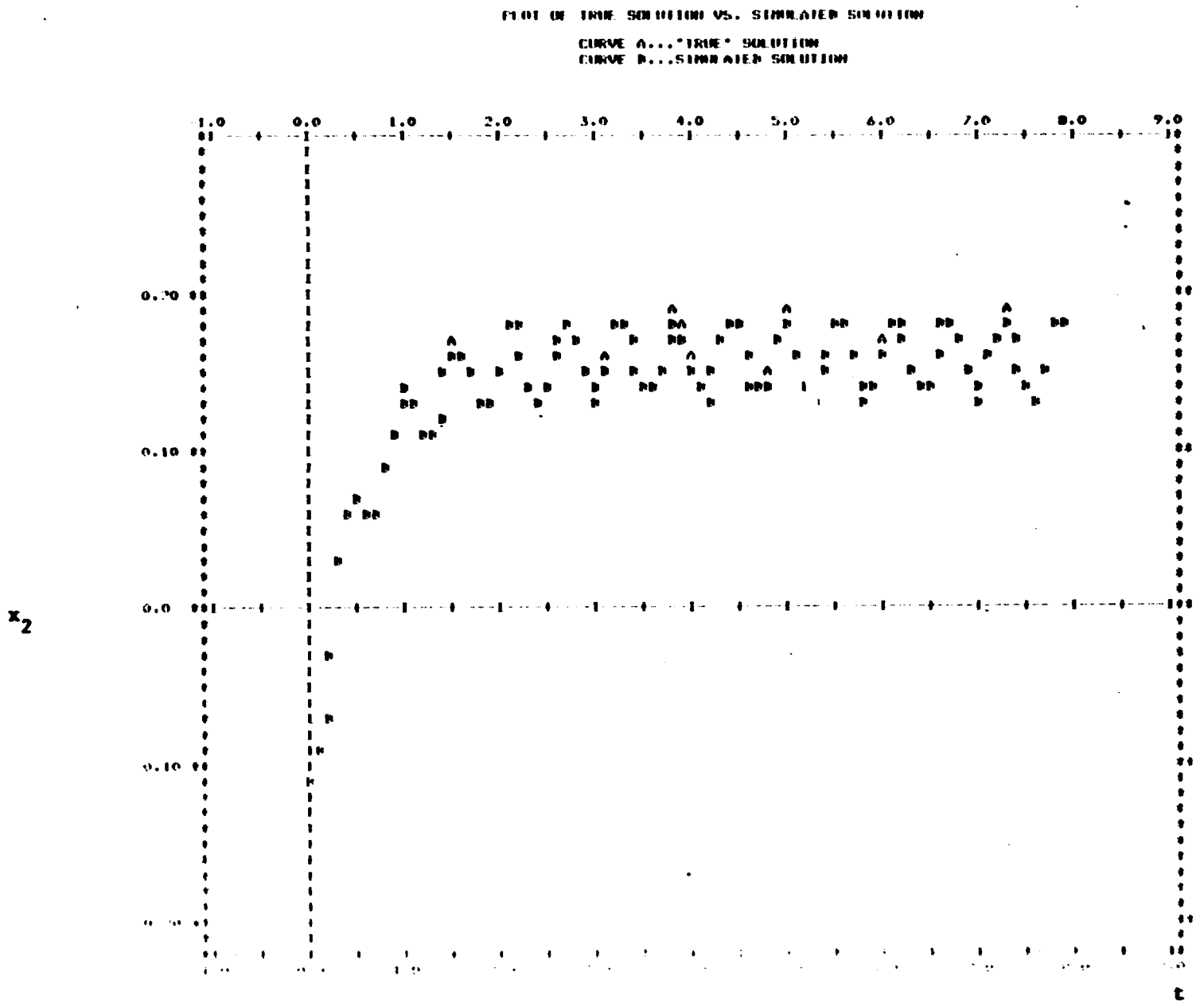

PLOT OF TRUE SOLUTION VS. SIMULATED SOLUTION
CURVE A...'TRUE' SOLUTION
CURVE B...SIMULATED SOLUTION
-0.50
0.0
0.50
1.00
1.50
2.00
2.50
3.00
3.50
4.00
4.50
0.080
0.060
0.040
0.020
0.0
$x_3$
t

Figure 2.27




$x_1$

Figure 2.28




$x_2$

Figure 2.29

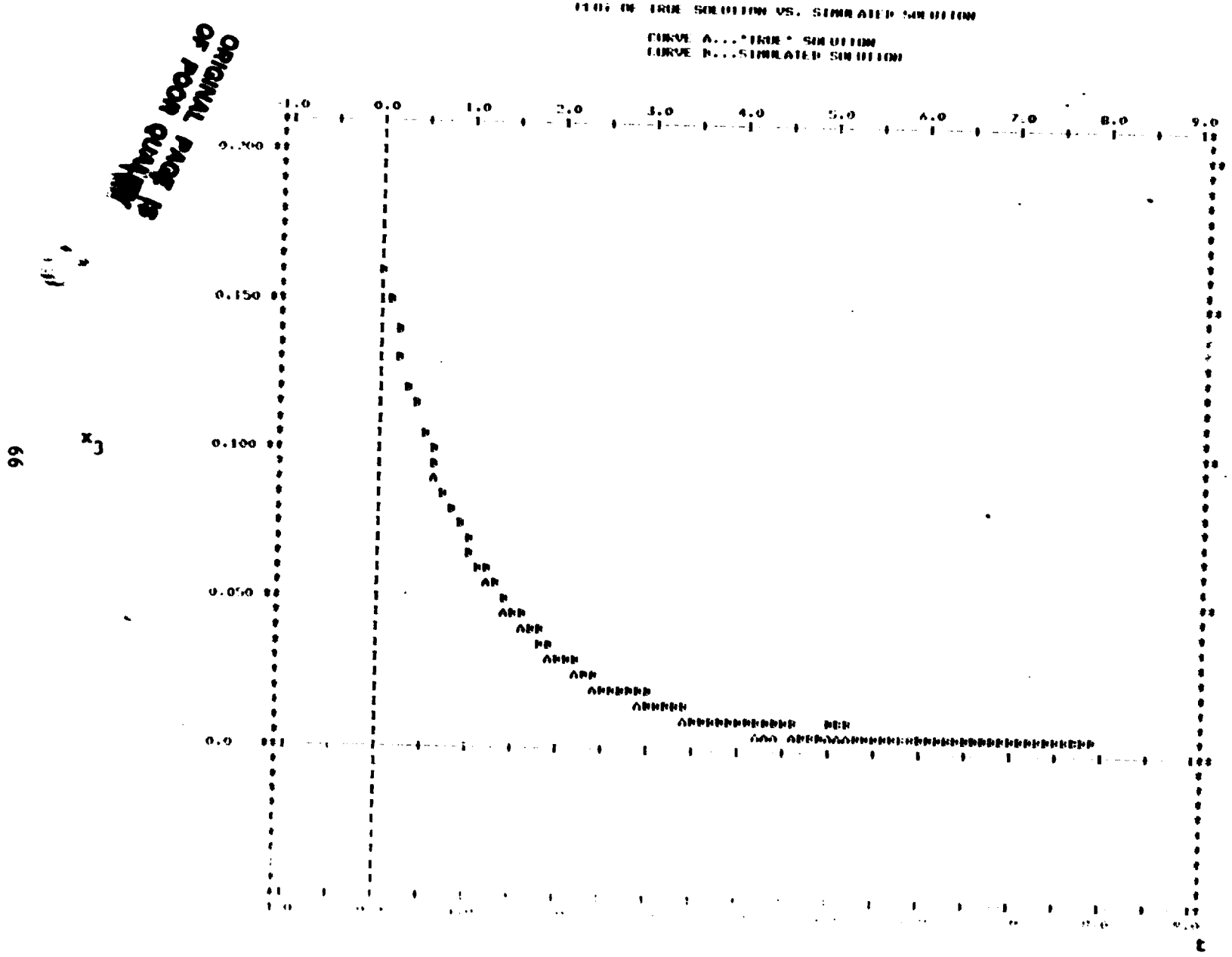



Figure 2.30

### 2.1.5 Conclusion

An important aspect of the identification scheme as put forth in this chapter concerns the capabilities of routine SIMEQUAT and the least squares technique in general. A question of conditioning arises as the size of the problem increases. Three factors contribute to problem size: the length of the state and input vectors; the degree of the approximation; and the number of sample points used in observing the system. For larger problems it is this last factor which in some sense limits the actual identification algorithm.

Some comments concerning actual computer time are in order. As an example, the identification for the third order example presented earlier was carried out using interactive software on a terminal over a time-sharing network, IBM system 370. The entire identification process took approximately ten minutes terminal time, corresponding to about five seconds of CPU time. The verifying simulations took almost as long, due to the size of the reconstructed equations. Smaller examples are even faster, although the above-mentioned times are certainly reasonable.

As a final note, it is interesting to consider use of other sampling techniques in the identification procedure. For instance, suppose that variable sample periods were used in an attempt to better observe the system. In fact, an apriori knowledge of particular behavior of the observed data could lead to clever choices for sample periods. The time between samples could even be generated randomly.

Results of the discussion and examples of this chapter indicate that the technique can be utilized in a number of applications. Given a block of observed data from a nonlinear system, and given knowledge of the number of states and inputs, several model systems (for different operating

regions) in the form of a collection of individual linear operators can be identified.

## 2.2 Report on CARDIAD Progress (R.M. Schafer)

Though funded work on the CARDIAD method was discontinued more than six months ago, its progress has continued steadily by means of fellowship support extended by the University of Notre Dame. During this period, the emphasis has been placed upon developing the method to be helpful with diagonal dominance designs in cases having four or more inputs and outputs.

Appendix B contains the preprint of an especially challenging design on such a model.

## 2.3 Report on Feedback Loop Closures (V. Seshadri)

Also mentioned in the last Semi-Annual Status Report was a recent extension of early grant work in the general area of pole and zero assignment and the exterior algebra. This work received support from Grant 3048 in its early stages, but in later years received assistance from different sources.

During the period of this report, another presentation has been made on this subject. The preprint is contained in Appendix C.

## APPENDIX A

## GRANT BIBLIOGRAPHY, INCEPTION TO PRESENT


(1) R.J. Leake, J.L. Melsa, and M.K. Sain, "Preliminary Proposal on Modern Approaches to Jet Engine Control", November 1, 1974.

(2) R.J. Leake, M.K. Sain, and J.L. Melsa, "Proposal to NASA for Support of a Work Entitled 'Alternatives for Jet Engine Control'", November 27, 1974.

(3) R.J. Leake and M.K. Sain, "Semi-Annual Status Report on NASA Grant NSG-3048, 'Alternatives for Jet Engine Control'", March 1, 1975 - August 31, 1975.

(4) R.J. Leake, M.K. Sain, and J.L. Melsa, "Proposal for Continuation of NASA Grant NSG-3048, 'Alternatives for Jet Engine Control'", October 1, 1975.

(5) J.C. Shearer, "An IBM 370/158 Installation and User's Guide for the DYNGEN Jet Engine Simulator", M.S. Thesis, November 1975.

(6) T.C. Brennan and R.J. Leake, "Simplified Simulation Models for Control Studies of Turbojet Engines", Technical Report EE 757, University of Notre Dame, November 1975.

(7) R.J. Leake, M.K. Sain, and J.L. Melsa, "Final Technical Report on NASA Grant NSG-3048, 'Alternatives for Jet Engine Control'", March 1, 1975 - February 29, 1976.

(8) R.R. Gejji and M.K. Sain, "Polynomial Techniques Applied to Multivariable Control of Jet Engines", Technical Report EE 761, March 1976.

(9) V. Seshadri and M.K. Sain, "Multivariable System Compensation Including Characteristic Methods for Jet Engine Control", Technical Report EE 763, May 1976.

(10) A.J. Maloney III, "Graphics Analysis of Dominance in Jet Engine Control Models", Technical Report EE 765, June 1976.

(11) M.K. Sain, R.J. Leake, R. Basso, R. Gejji, A. Maloney, and V. Seshadri, "Alternative Methods for the Design of Jet Engine Control Systems", Proceedings Fifteenth Joint Automatic Control Conference, pp. 133-142, July 1976.

(12) J.C. Shearer, W.E. Longenbaker, Jr., and R.J. Leake, "An IBM 370/158 Installation and User's Guide to the DYNGEN Jet Engine Simulator", Technical Report EE 766, July 1976.

(13) R.R. Gejji and M.K. Sain, "A Jet Engine Control Problem for Evaluating Minimal Design Software", Proceedings Midwest Symposium on Circuits and Systems, pp. 238-243, August 1976.

(14) V. Seshadri and M.K. Sain, "Interaction Studies on a Jet Engine Model by Characteristic Methodologies", Proceedings Midwest Symposium on Circuits and Systems, pp. 232-237, August 1976.

(15) R.J. Leake and M.K. Sain, "Semi-Annual Status Report on NASA Grant NSG-3048, 'Alternatives for Jet Engine Control', Supplement No. 1", March 1, 1976-August 31, 1976.

(16) R. Basso and R.J. Leake, "Computational Alternatives to Obtain Time-Optimal Jet Engine Control", Technical Report EE 767, September 1976.

(17) R. Basso and R.J. Leake, "Computational Methods to Obtain Time-Optimal Jet Engine Control", Proceedings Fourteenth Allerton Conference on Circuit and System Theory, pp. 652-661, September 1976.

(18) M.K. Sain and V. Seshadri, "A Result on Pole Assignment by Exterior Algebra", Proceedings Fourteenth Allerton Conference on Circuit and System Theory, pp. 399-407, September 1976.

(19) R.R. Gejji, "A Computer Program to Find the Kernel of a Polynomial Operator", Proceedings Fourteenth Allerton Conference on Circuit and System Theory, pp. 1091-1100, September 1976.

(20) R.J. Leake and M.K. Sain, "Proposal for Continuation of NASA Grant NSG-3048, 'Alternatives for Jet Engine Control'", October 25, 1976.

(21) R.M. Schafer, W.E. Longenbaker, and M.K. Sain, "System Dominance: A Preliminary Report on an Alternate Approach", Technical Report, University of Notre Dame, November 25, 1976.

(22) R.J. Leake and M.K. Sain, "Final Technical Report, NASA Grant NSG-3048, 'Alternatives for Jet Engine Control', Supplement No. 1", March 1, 1976 - February 28, 1977.

(23) R.J. Leake, J.G. Allen, and R.S. Schlunt, "Optimal Regulators and Follow-Up Systems Determined from Input-Output Signal Monitoring", Proceedings IEEE International Symposium on Circuits and Systems, April 1977.

(24) R.M. Schafer, "A Graphical Approach to System Dominance", Technical Report EE 772, University of Notre Dame, April 1, 1977.

(25) W.E. Longenbaker and R.J. Leake, "Hierarchy of Simulation Models for a Turbofan Gas Engine", Proceedings Eighth Annual Pittsburgh Conference on Modeling and Simulation, April 1977.

(26) P.W. Hoppner, "The Direct Approach to Compensation of Multivariable

Jet Engine Models", Technical Report EE 774, University of Notre Dame, May 1977.

(27) R.R. Gejji and R.J. Leake, "Time-Optimal Control of a Single Spool Turbojet Engine Using a Linear Affine Model", Technical Report EE 7711, University of Notre Dame, June 1977.

(28) R.M. Schafer, R.R. Gejji, P.W. Hoppner, W.E. Longenbaker, and M.K. Sain, "Frequency Domain Compensation of a DYNGEN Turbofan Engine Model", Proceedings Sixteenth Joint Automatic Control Conference, pp. 1013-1018, June 1977.

(29) R.R. Gejji and M.K. Sain, "Application of Polynomial Techniques to Multivariable Control of Jet Engines", Proceedings Fourth IFAC Symposium on Multivariable Technological Systems, pp. 421-429, July 1977.

(30) R.R. Gejji, "Use of DYGABCD Program at Off-Design Points", Technical Report EE 7703, University of Notre Dame, July 1977.

(31) E.A. Sheridan and R.J. Leake, "Non-Interactive State Request Jet Engine Control with Non-Singular B Matrix", Proceedings Twentieth Midwest Symposium on Circuits and Systems, pp. 539-543, August 1977.

(32) R. Gejji, R.M. Schafer, M.K. Sain, and P. Hoppner, "A Comparison of Frequency Domain Techniques for Jet Engine Control System Design", Proceedings Twentieth Midwest Symposium on Circuits and Systems, pp. 680-685, August 1977.

(33) W.E. Longenbaker and R.J. Leake, "Time Optimal Control of a Two-Spool Turbofan Jet Engine", Technical Report EE 7714, University of Notre Dame, September 1977.

(34) R.J. Leake and J.G. Comiskey, "A Direct Method for Obtaining Nonlinear Analytical Models of a Jet Engine", Proceedings International Forum on Alternatives for Linear Multivariable Control, National Electronics Conference, Chicago, pp. 203-212, October 1977.

(35) M.K. Sain and R.J. Leake, "Proposal for Continuation of NASA Grant NSG-3048, 'Alternatives for Jet Engine Control'", October 25, 1977.

(36) J.A. Ortega and R.J. Leake, "Discrete Maximum Principle with State Constrained Control", SIAM Journal on Control and Optimization, Vol. 15, No. 6, pp. 109-115, November 1977.

(37) Michael K. Sain and V. Seshadri, "Pole Assignment and a Theorem from Exterior Algebra", Proceedings IEEE Conference on Decision and Control, pp. 291-295, December 1977.

(38) R. Michael Schafer and Michael K. Sain, "Some Features of CARDIAD Plots for System Dominance", Proceedings IEEE Conference on Decision

and Control, pp. 801-806, December 1977.

(39) M.K. Sain, "The Theme Problem", in Alternatives for Linear Multivariable Control, M.K. Sain, J.L. Peczkowski and J.L. Melsa, Editors. Chicago: National Engineering Consortium, 1978, pp. 20-30.

(40) R.M. Schafer and M.K. Sain, "Input Compensation for Dominance of Turbofan Models", in Alternatives for Linear Multivariable Control, M.K. Sain, J.L. Peczkowski, and J.L. Melsa, Editors. Chicago: National Engineering Consortium, 1978, pp. 156-169.

(41) J.L. Peczkowski and M.K. Sain, "Linear Multivariable Synthesis with Transfer Functions", in Alternatives for Linear Multivariable Control, M.K. Sain, J.L. Peczkowski, and J.L. Melsa, Editors. Chicago: National Engineering Consortium, 1978, pp. 71-87.

(42) R.J. Leake and M.K. Sain, "Semi-Annual Status Report, NASA Grant NSG-3048, 'Alternatives for Jet Engine Control', Supplement No. 2", March 1, 1977 - August 31, 1977.

(43) R.J. Leake and M.K. Sain, "Final Technical Report, NASA Grant NSG-3048, 'Alternatives for Jet Engine Control', Supplement No. 2", March 1, 1977 - February 28, 1978.

(44) M.K. Sain and R.J. Leake, "Semi-Annual Status Report, NASA Grant NSG-3048, 'Alternatives for Jet Engine Control'", March 1, 1978 - August 31, 1978.

(45) R.J. Leake, J.L. Peczkowski, and M.K. Sain, "Step Trackable Linear Multivariable Plants", Technical Report EE 789, September 1978.

(46) R.M. Schafer and M.K. Sain, "CARDIAD Design: Progress in the Four Input/Output Case", Proceedings Sixteenth Allerton Conference on Communication, Control, and Computing, p. 567, October 1978.

(47) M.K. Sain, "Proposal for Continuation of NASA Grant NSG-3048, 'Alternatives for Jet Engine Control'", November 1, 1978.

(48) M.K. Sain, "On Exosubsets and Internal Models", Proceedings IEEE Conference on Decision and Control, pp. 1069-1073, January 1979.

(49) M.K. Sain, "Annual Technical Report, NASA Grant NSG-3048, 'Alternatives for Jet Engine Control'", March 1, 1978 - February 28, 1979.

(50) V. Seshadri and M.K. Sain, "An Application of Exterior Algebra to Multivariable Feedback Loops", Proceedings Conference on Information Sciences and Systems, Johns Hopkins University, pp. 337-342, 1979.

(51) M.K. Sain and R.M. Schafer, "Alternatives for Jet Engine Control", Propulsion Controls Symposium, NASA Lewis Research Center, Preprints, p. III-15, May 1979.

(52) J.G. Comiskey, "Time Optimal Control of a Jet Engine Using a Quasi-Hermite Interpolation Model", Technical Report EE 791, May 1979.

(53) R.M. Schafer and M.K. Sain, "Frequency Dependent Precompensation for Dominance in a Four Input/Output Theme Problem Model", Proceedings Eighteenth Joint Automatic Control Conference, pp. 348-353, June 1979.

(54) M.K. Sain, "Semi-Annual Status Report , NASA Grant NSG-3048, 'Alternatives for Jet Engine Control'", March 1, 1979-August 31, 1979.

(55) V. Seshadri and M.K. Sain, "Loop Closures and the Induced Exterior Map", Proceedings Seventeenth Allerton Conference on Communications, Control, and Computing, pp. 753-761, October 1979.

(56) R. Michael Schafer and Michael K. Sain, "CARDIAD Approach to System Dominance with Application to Turbofan Engine Models", Proceedings Thirteenth Asilomar Conference on Circuits, Systems, and Computers, November 1979.

(57) R.J. Leake, J.L. Peczkowski, and M.K. Sain, "Step Trackable Linear Multivariable Plants", International Journal of Control, Vol. 30, pp. 1013-1022, December 1979.

(58) M.K. Sain, "Semi-Annual Status Report, NASA Grant NSG-3048, 'Alternatives for Jet Engine Control'", September 1, 1979-February 29, 1980.

(59) M.K. Sain, "Proposal for Continuation of NASA Grant NSG-3048, 'Alternatives for Jet Engine Control'", April 15, 1980.

APPENDIX B

"CARDIAD APPROACH TO SYSTEM DOMINANCE

WITH APPLICATION TO TURBOFAN ENGINE MODELS"

R.M. Schafer and M.K. Sain

Asilomar Conference

November 1979

# CARDIAD APPROACH TO SYSTEM DOMINANCE WITH APPLICATION TO TURBOFAN ENGINE MODELS*

R. Michael Schafer and Michael K. Sain
Department of Electrical Engineering
University of Notre Dame
Notre Dame, Indiana
USA 46556



## Summary

One of the features of present day research on linear multivariable systems has been a renewed interest in Nyquist methods. Based upon the determinant of return difference, these methods must develop procedures which interface with skew symmetric, multilinear forms. A well known interface has been made by Rosenbrock, who used a concept called diagonal dominance. This paper reports on a graphical, interactive way to achieve the concept.

## Introduction

In recent years, increased attention has been paid to the use of frequency domain techniques for the design of multivariable control systems. Most of these techniques are based upon the equation

$$p_c(s) = |M(s)|\, p_0(s) \qquad (1)$$

which relates the zeros of the open loop characteristic polynomial $p_0(s)$ and the zeros of the closed loop characteristic polynomial $p_c(s)$ through the determinant of the return difference matrix $M(s)$. Given that the zeros of the open loop characteristic polynomial are known, stability of the closed loop characteristic polynomial can be determined by Nyquist analysis of $|M(s)|$. Unfortunately, direct Nyquist analysis of $|M(s)|$ yields little design insight. Therefore, alternate means of studying $|M(s)|$ have been devised.

In the Inverse Nyquist Array approach due to Rosenbrock, the system is first compensated to achieve diagonal dominance. An $n \times n$ matrix $Z(s)$ is said to be diagonally column dominant if for all $s$ on the Nyquist contour $D$, and $i = 1,2,\ldots,n$,

$$|z_{ii}(s)| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |z_{ji}(s)|. \qquad (2)$$

If this condition is satisfied, the usual net encirclements made by the Nyquist plot of $|M(s)|$ are equal to the sum of the net encirclements made by the diagonal entries of $M(s)$. Thus, stability can be determined by Nyquist analysis of the diagonal entries of $M(s)$.

The CARDIAD (Complex Acceptability Region for DIAgonal Dominance) method is a graphical technique for achieving this dominance condition.

## CARDIAD Method

Consider the system of Figure 1. For the purposes of this paper, $G(s)$ represents a 4-input, 4-output model of a turbofan jet engine. It is desired to design the compensator $K(s)$ such that $G(s)\,K(s)$ is column dominant. The compensator is normalized to having 1's on the main diagonal, so that dominance is achieved in a given column of $G(s)\,K(s)$ by appropriate choice of the off-diagonal entries in the corresponding column of $K(s)$.

At a frequency, a sufficient condition for dominance can be expressed in a quadratic inequality of the form

$$f(x) = x^t A x + x^t b + c > 0, \qquad (3)$$

where $A$, $b$, and $c$ are respectively a matrix, a vector, and a scalar formed by evaluation of the plant transfer function matrix at the frequency being studied, where $x$ is a vector of the real and imaginary parts of the off-diagonal entries of a column of the compensator, and where superscript $t$ denotes transpose. Dominance is achieved by choosing $x$ such that $f(x)$ is positive.

Several approaches are used to choose $x$ such that $f(x) > 0$. Since it is desirable to achieve dominance with as simple a compensator as possible, the gradient of $f(x)$ is taken with respect to each entry $x_i$ assuming all other entries are zero. Here, $x_i$ may be understood as a pair $(r_i, i_i)$ consisting of the real and imaginary parts of some off-diagonal compensator entry. This approach, referred to as type 1 analysis, attempts to achieve dominance in a column by using only one nonzero, off-diagonal entry for the column of the compensator. In the event that it is impossible to achieve dominance with only one nonzero, off-diagonal entry, the gradient of $f(x)$ with respect to all variables is taken. This approach is referred to as type 2 analysis and utilizes all off-diagonal entries of the compensator to achieve dominance in a column. A third means of choosing the vector $x$ is used in the event that the hessian in the type 2 analysis approach is indefinite. It is known that one solution to making $G(s)$ dominant is to compensate with the inverse system. Thus, a solution for the vector $x$ is to choose the values of the inverse system at that frequency, normalized to 1 on the diagonal so as to fit the form of the compensator $K(s)$. In the case where the hessian is negative definite, this inverse system analysis, known as type 4 analysis, predicts the same solution as the type 2 analysis plots.

The CARDIAD plot is a graphical representation of the results of the gradient analysis. Consider type 1 analysis of a given column. $f(0,\ldots 0,x_i,0..)$ is a paraboloid in 3-space, and the value found by the gradient analysis can be a positive maximum, a negative maximum, a positive minimum, or a negative minimum. In the positive maximum case, any value of $x_i$ which lies inside the intersection of $f(x)$ and the complex plane $x_i$ will make $f(..0,x_i,0,..)$ positive; and dominance will be achieved at the frequency being studied. In the CARDIAD plot, this is represented by a solid circle which is the solution of $f(..0,x_i,0..) = 0$, and a '+' at the value of $x_i$ where the gradient vanishes, which is at the center of the circle. In the case of a negative minimum, all values of $x_i$ lying outside the circle $f(...0,x_i,0..) = 0$ will make $f(..0,x_i,0...)$ positive. In this case, an 'x' is drawn at the value where the gradient vanishes and a dashed circle at $f(...0,x_i,0..) = 0$. In the negative maximum case, no value of $x_i$ will achieve dominance; and a 'A' is drawn. In the positive minimum case, any value of $x_i$

*This work has been supported in part by the National Aeronautics and Space Administration under Grant NSG-3048.

will achieve dominance; in the column at this frequency, and a '□' is drawn.

In types 2 and 4 analyses, the center symbols are drawn at the gradient values, but the center type and circle type are decided by making a worst case deviation from the gradient values of all but one of the entries of z; and then the remaining entry is analyzed in a fashion analogous to type 1 analysis.

A CARDIAD plot results when this graphical gradient information is plotted over a range of frequencies. Figures 2 and 3 are typical CARDIAD plots and will be used to describe compensator design.

Figure 2 is a type 1 analysis plot which contains only solid circles. In this case, there exist constant real values $(x_1,0)$ for $x_1$ which lie inside all of the solid circles. Hence, to achieve dominance in this column at all frequencies, any such choice of $x_1$ will suffice, since $f(..0,x_1,0..)$ will then be positive at all frequencies. In Figure 3, there exists no such constant real value, but a simple first order entry which as a function of frequency traces the centers of the circles can be used. Thus, if the CARDIAD plot indicates that no constant real value will achieve dominance, the shape of the plot guides the designer in determing a frequency dependent entry.

## Design Example

The model used in the following design example is taken from [1]. It is a sixth order, 4-input, 4-output description of a turbofan engine.

As a first step in the design procedure, the model was compensated with the inverse system evaluated at $s = 0$. Figures 4-8 are the type 1 analysis plots of the 4,2 entry, the 3,4 entry, and the entire first column. Type 1 analysis of the first column indicates that dominance cannot be achieved using only one nonzero, off-diagonal entry. The same was true for the third column. Figures 9-11 are the type 4 analysis plots for the first column; and Figures 12-14 are the type 2 analysis plots for the third column.

In both the second column and the fourth column, dominance was achievable using type 1 analysis and constant compensation as described in the discussion of Figure 2. Dominance was achieved in column 2 by choosing the 4,2 entry to be -880.8. Note that this value lies within all solid circles and outside all dashed circles. In like manner, the fourth column was made dominant by choosing the 3,4 entry to be -.59.

In the first and third columns, it was necessary to fit all three off-diagonal entries of the compensator to the shape of the centers of the type 4 and 2 plots, respectively. In each case, second order compensation was necessary to fit adequately the shapes. The three entries chosen for the first column were

$$k_{2,1}(s) = \frac{-.189E\text{-}3s^2 - .0129s}{.227E\text{-}2s^2 + .238s + 1},$$

$$k_{3,1}(s) = \frac{-.044s^2 - 2.30s}{.227E\text{-}2s^2 + .238s + 1},$$

$$k_{4,1}(s) = \frac{.146s^2 + 3.56s}{.227E\text{-}2s^2 + .238s + 1}.$$

The third column was made dominant with the following three off-diagonal entries

$$k_{1,3}(s) = \frac{-.175E\text{-}4s^2 + .263E\text{-}2s}{.198E\text{-}2s^2 + .0837s + 1},$$

$$k_{2,3}(s) = \frac{.333E\text{-}5s^2 - .274E\text{-}3s}{.198E\text{-}2s^2 + .0837s + 1},$$

$$k_{4,3}(s) = \frac{-.635E\text{-}2s^2 - .0188s}{.198E\text{-}2s^2 + .0837s + 1}.$$

It should be noted that, in each case, the denominator polynomial of the column is the same, thereby keeping the order of the resulting compensator small.

With compensation as described above, type 1 analysis was repeated to verify that dominance has been achieved. Figures 15-18 are a type 1 analysis plot from each column. Note that in every case, the plot predicts that an acceptable solution is the origin. Since type 1 analysis is drawn assuming all other off-diagonal entries are zero, this implies that the columns are now dominant, since the plots predict that identity compensation will achieve dominance.

## Discussion

The tools associated with Nyquist analysis of (1) are often helpful in the analysis and design of multivariable systems in the frequency domain. One of the ways to approach the design of $|M(s)|$ in (1) is by means of the dominance ideas of Rosenbrock [2]. This paper describes a graphical, interactive procedure for attaining dominance. For other examples, see [3].

## References


1. E.A. Spang III, "Insight into the Application of the Inverse Nyquist Array Method to Turbofan Engine Control", in Alternatives for Linear Multivariable Control, M.K. Sain, J.L. Peczkowski, and J.L. Melsa, eds., 1978, pp. 138-155.

2. H.H. Rosenbrock and N. Munro, "The Inverse Nyquist Array Method", ibid., pp. 101-137.

3. R.M. Schafer and M.K. Sain, "Input Compensation for Dominance of Turbofan Models", ibid., pp. 156-169.

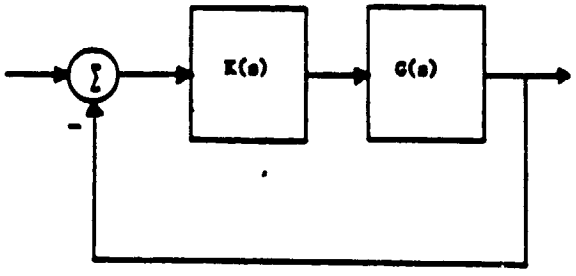



Figure 1

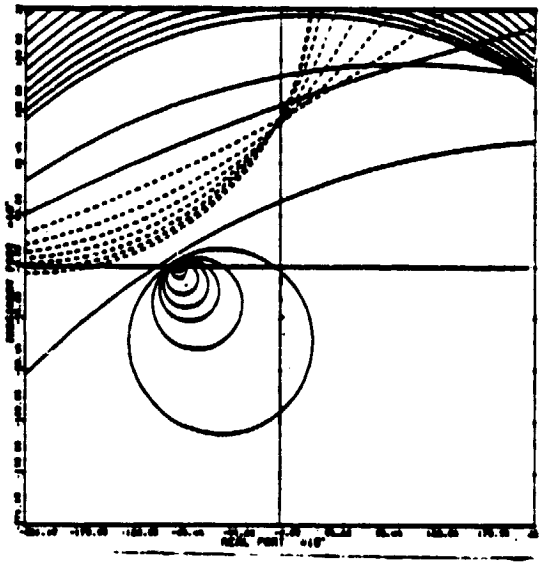


Figure 4

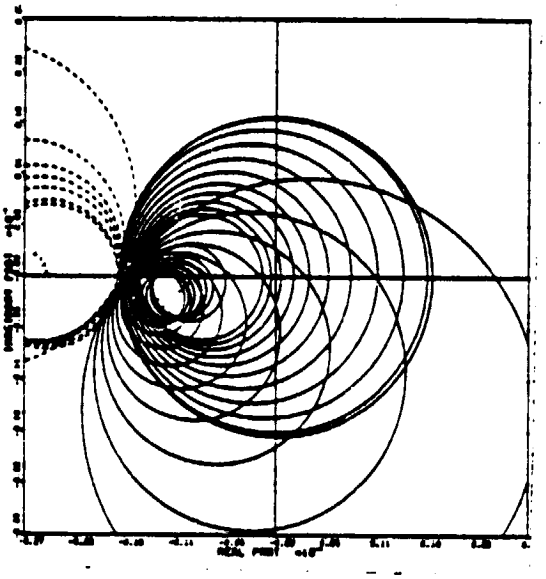


Figure 2

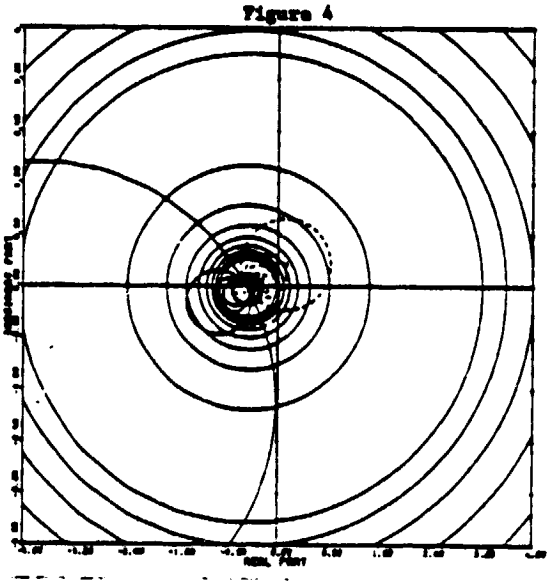


Figure 5

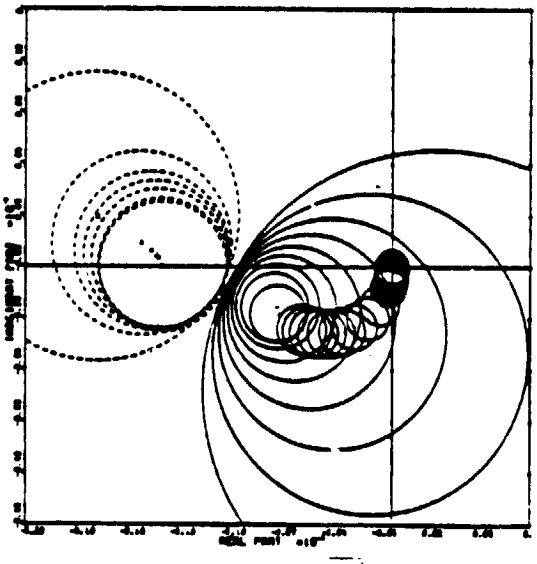


Figure 3

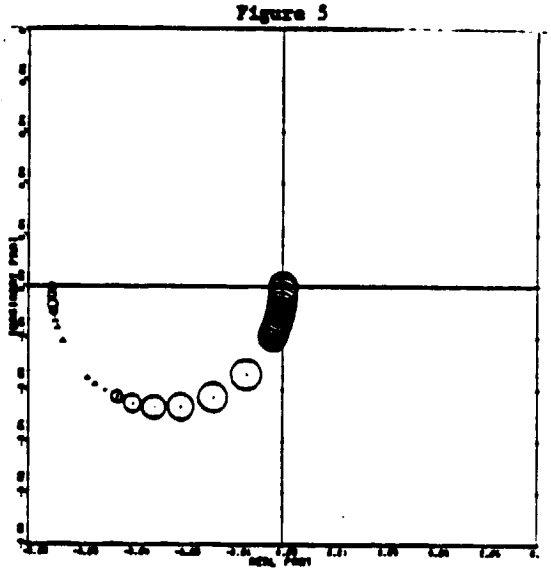


Figure 6

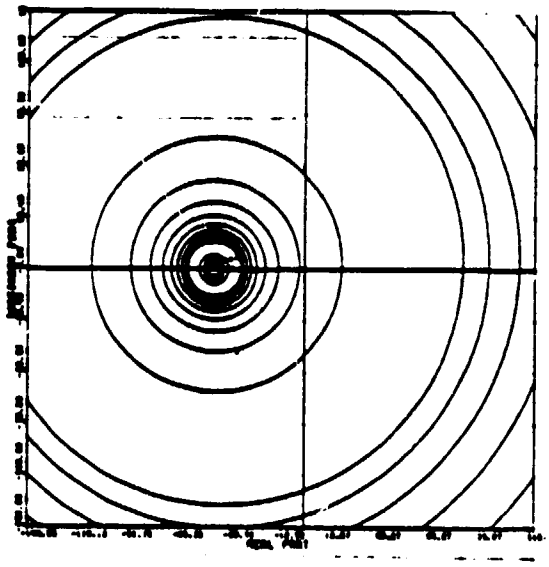


Figure 7

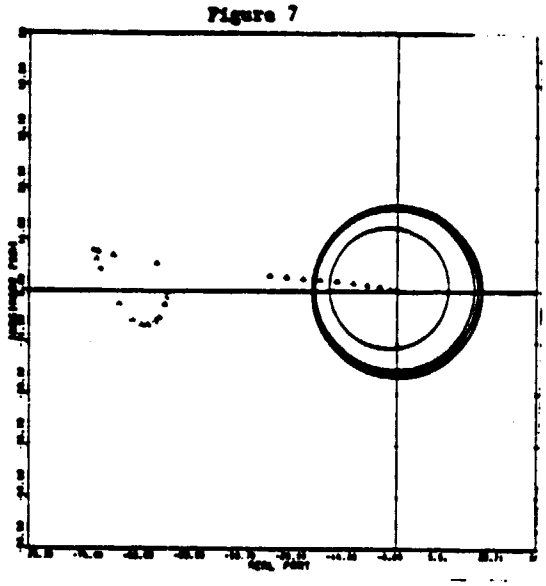


Figure 8

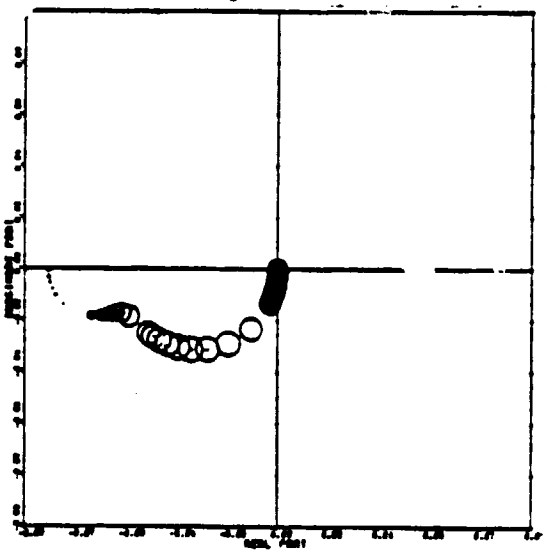


Figure 9

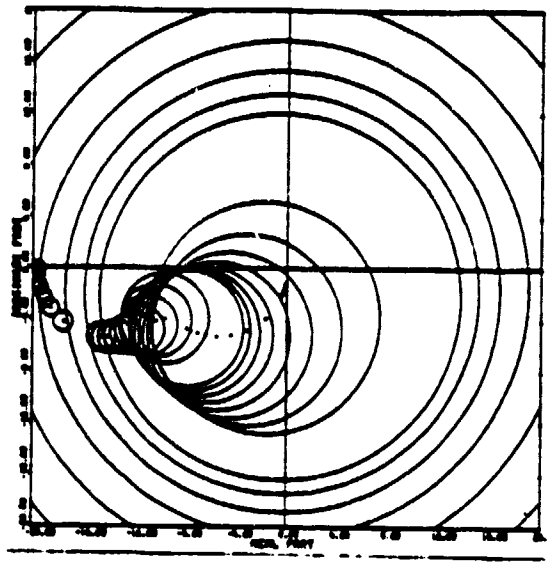


Figure 10

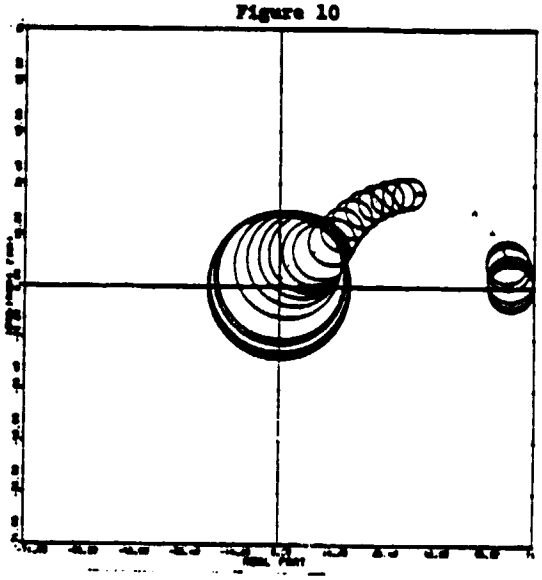


Figure 11

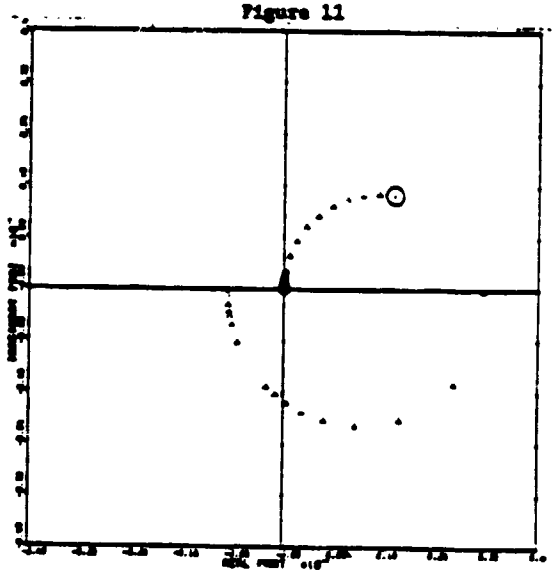


Figure 12

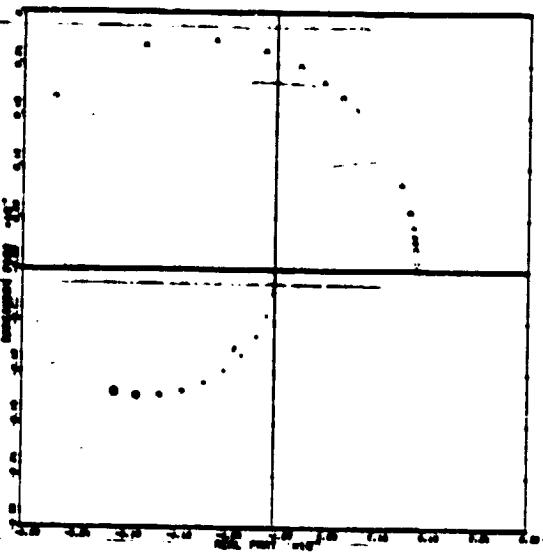


Figure 13

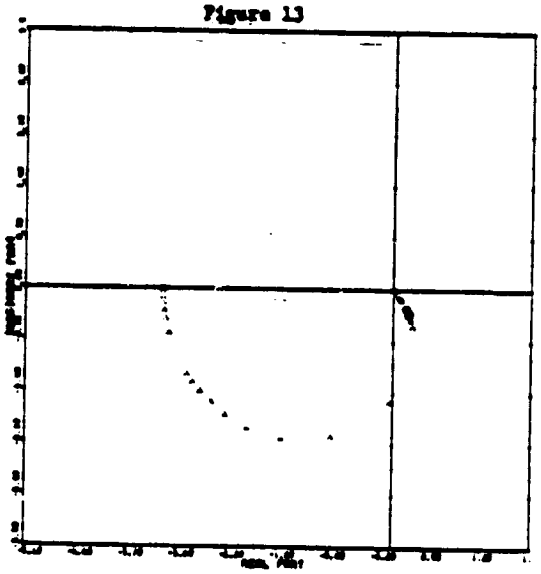


Figure 14

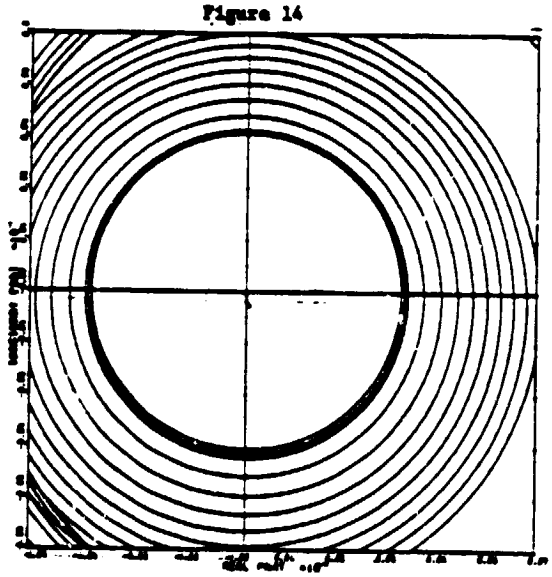


Figure 15

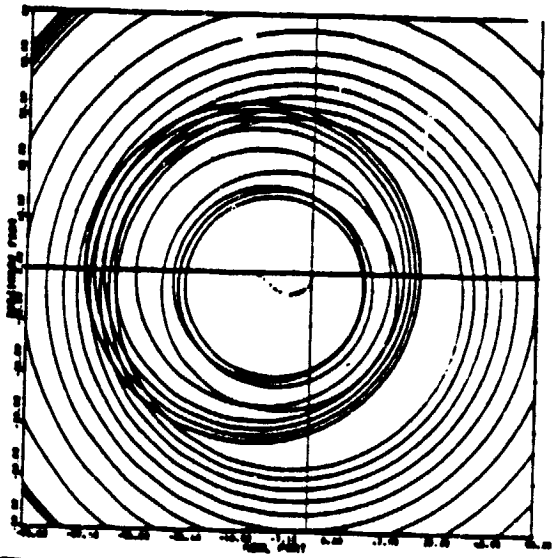


Figure 16

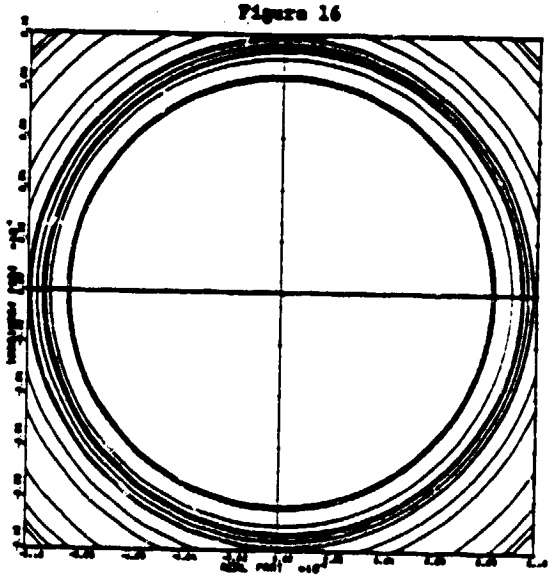


Figure 17

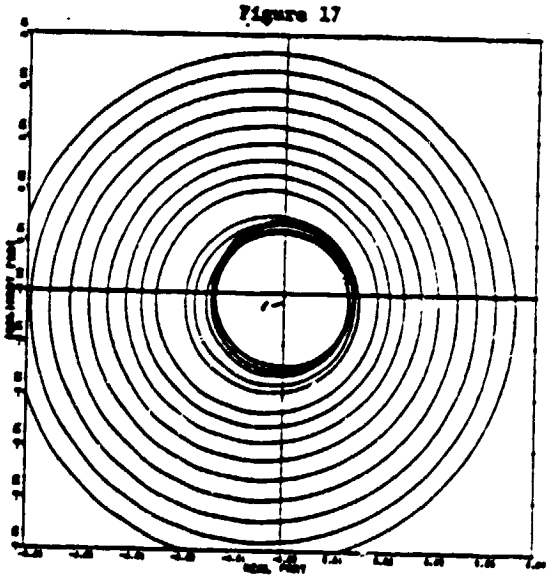


Figure 18

APPENDIX C

"LOOP CLOSURES AND THE INDUCED EXTERIOR MAP"

V. Seshadri and M.K. Sain

Allerton Conference

October 1979

# LOOP CLOSURES AND THE INDUCED EXTERIOR MAP*

V. SESHADRI
Dept. Electrical Engineering
General Motors Institute
Flint, Michigan
USA 48502

M.K. SAIN
Dept. Electrical Engineering
University of Notre Dame
Notre Dame, Indiana
USA 46556

## ABSTRACT

In this paper we establish the connection between linear multivariable feedback loop closures and the induced exterior map, the latter being the by-product of exterior algebras over the input and output vector spaces. This suggests the concept of a sequence of multivariable zeros which should enhance the designer's ability to shape multi-output transients.

## Introduction

Much interest has been evinced in recent years about the exterior algebra, a structure especially suited for addressing questions related to matrix determinants and inverses. It has been shown that this structure can be used to solve problems in pole assignment [1,2], in individual zero placement [3] and indeed in a host of areas related to systems and information theory [4]. The present paper may roughly be divided into three parts. The first part introduces the exterior algebra. The presentation is extremely brief due to limitations of space; for more details the reader is referred to Greub [5]. The second part of the paper considers multivariable feedback loop closures, the loops being closed one by one, and studies their relation to certain entities called numerators of the $k$th kind, the latter arising from a frequency design method used in industry [6]. The last part of the paper establishes the connection between numerators of the $k$th kind and the $k$th induced exterior map, the latter being a by-product of exterior algebras over the input and output vector spaces [7]. In this way we will see that when considering multivariable feedback loop closures it would be helpful to view them in the context of the appropriate induced exterior morphism.

## The Exterior Algebra

Consider an F-vector space $V$. We can construct an exterior algebra [5] $\Lambda V$ over $V$. The bilinear operator introduced by this construction is commonly called the exterior product or the "wedge" product $\wedge$, and operates as

$$(\alpha_1 a_1 + \alpha_2 a_2) \wedge a_3 = \alpha_1 a_1 \wedge a_3 + \alpha_2 a_2 \wedge a_3$$

$$a_1 \wedge (\alpha_3 a_3 + \alpha_4 a_4) = \alpha_3 a_1 \wedge a_3 + \alpha_4 a_1 \wedge a_4,$$

where $a_1, a_2, a_3, a_4$ belong to the algebra $\Lambda V$, and $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ are field elements from F. Furthermore, the operator $\wedge$ is skew-symmetric.

Now consider a map $f : V \to W$. If we construct the exterior algebras $\Lambda V$ and $\Lambda W$ over the vector spaces $V$ and $W$ respectively, the map $f$ induces a morphism $f^\wedge$ over the algebras [7], which is just a sequence of

*This work has been supported in part by the National Aeronautics and Space Administration under Grant NSG-3048 and in part by the Office of Naval Research under contract N00014-79-C-0475.

maps $f_k^\wedge$ over the kth exterior spaces, as shown in Figure 1.

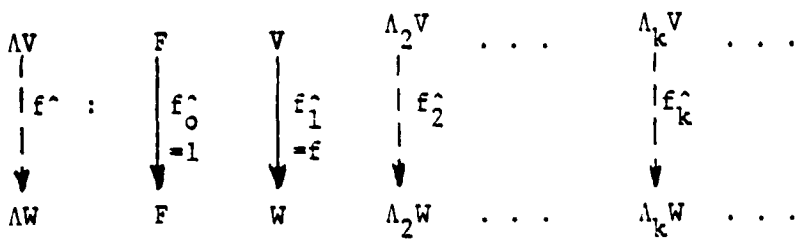



Figure 1. The Induced Exterior Morphism, $f^\wedge$.

A reasonable question that could be asked at this juncture is what relevance the induced exterior morphism $f^\wedge$ has to multivariable feedback control design. The answer is that there already exists a feedback control design method in industry [6] that makes partial use of the induced exterior morphism structure. It is not clear, however, that the originators of the method are aware of the structure and it appears that the design method could be extended by making fuller use of the sequence of exterior maps. In order to address these issues, let us first consider a problem suggested by Hofmann, et al. [6] and study the connection between feedback loop closures and certain entities called numerators of various kinds, the latter arising when the following problem is solved.

Problem: Obtain expressions for the arbitrary closed-loop transfer function $y_c/r_d$ after feeding back one or more outputs (measurements) to one or more inputs (controls), for the plant shown in Figure 2.

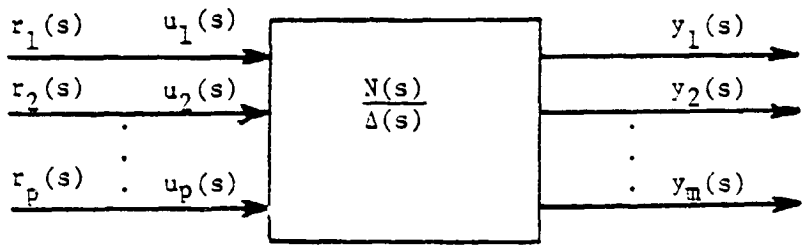



Figure 2. A Feedback Loop Closure Problem.

Comments: Figure 2 shows a multi-input multi-output plant; it has p inputs and m outputs. The reference, input and output may be compactly called r, u and y, and considered as elements of R(s)-vector spaces R, U and Y respectively, R(s) being the field of ratios of polynomials in s with real coefficients, the denominator being non-zero. The plant is nth order, linear and stationary. The expression for the plant in Figure 2, that is $N(s)/\Delta(s)$, is considered to have been derived, with the usual assumption of zero initial conditions, from an input-output plant description in the s-domain written as

$$A(s)\ y(s) = B(s)\ u(s),$$

where y belongs to the R(s)-vector space of dimension m, $[R(s)]^m$, and u belongs to $[R(s)]^p$. In the following discussion we shall drop the explicit dependence of the variables on s, for notational convenience.

The outputs y may be expressed explicitly in terms of the inputs u as

$$y = A^{-1} B u$$

$$= \frac{N}{\Delta} u ,$$

where $\Delta$ is the nth degree characteristic polynomial for the plant. If A and B are left coprime,

$$\Delta = \det A.$$

N is represented in numerical calculations by an mxp matrix whose elements belong to R[s], that is, they are polynomials in s with real coefficients; N is the adjusted plant transfer function numerator matrix.

Solution: In the following treatment let the polynomial matrices A and B be expressed in terms of m-length column vectors as

$$A \triangleq [a_1 \; a_2 \; \ldots \; a_m] \quad , \quad B \triangleq [b_1 \; b_2 \; \ldots \; b_p].$$

Consider the case of a single feedback loop, that is, one output $y_i$ is fed back to the jth comparison point through a feedback gain $g_{ji}$, as shown in Figure 3.

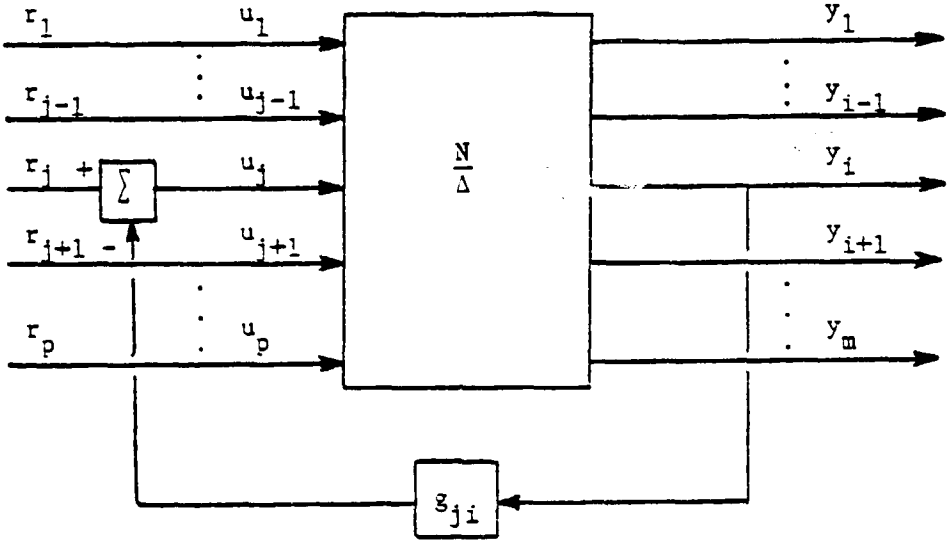



Figure 3. The Feedback Problem With One Loop Closure.

The relationship between the output y and the reference r may be expressed as

$$(A + BG)y = Br,$$

where y and u are given by

$$y = (y_1 \; \ldots \; y_m)^T \quad , \quad r = (r_1 \; \ldots \; r_p)^T.$$

G is a pxm matrix of zeros except for a single non-zero element $g_{ji} \in$ R(s), and is expressed in terms of p-length vectors in the same fashion as A and B above by

$$G = [0 \; \ldots \; 0 \; g_i \; 0 \; \ldots \; 0]$$

where

$$g_i = (0 \; .. \; 0 \; g_{ji} \; 0 \; .. \; 0)^T.$$

Note that $g_{ji}$ is the jth element in $g_i$, and that $g_i$ is the ith column of G. We can then express the equation

$$Ay = Bu$$

as

$$a_1y_1 + \ldots + a_cy_c + \ldots + a_{i-1}y_{i-1} + (a_i + b_jg_{ji})y_i + a_{i+1}y_{i+1} + \ldots + a_my_m$$
$$= b_1r_1 + \ldots + b_dr_d + \ldots + b_pr_p.$$

Note that in the above equation $a_k$, $k = 1,..,m$ and $b_k$, $k = 1,...,p$ are vectors whereas $y_k$, $k = 1,...,m$ and $r_k$, $k = 1,...,p$ are scalars, the field being $R(s)$.

We wish to isolate the closed-loop transfer function $y_c/r_d$; this may be achieved by taking the exterior product of both sides of the above equation with the (m-1)-exterior term

$$a_1 \wedge \ldots \wedge a_{c-1} \wedge a_{c+1} \wedge \ldots \wedge a_{i-1} \wedge (a_i + b_jg_{ji}) \wedge a_{i+1} \wedge \ldots \wedge a_m,$$

which is the exterior product of all the vectors on the LHS of the equation except for $a_c$, whose coefficient $y_c$ is the output of interest in our current discussion. Because of the multilinearity and skew-symmetry of the exterior product, all the terms on the LHS of the equation will become zero except for one term which includes $a_cy_c$. If, at the same time, all the references except the one of interest, $r_d$, are held zero, the equation becomes

$$a_1 \wedge \ldots \wedge a_{c-1} \wedge a_{c+1} \wedge \ldots \wedge a_{i-1} \wedge (a_i + b_jg_{ji}) \wedge a_{i+1} \wedge \ldots \wedge a_m \wedge a_cy_c$$
$$= a_1 \wedge \ldots \wedge a_{c-1} \wedge a_{c+1} \wedge \ldots \wedge a_{i-1} \wedge (a_i + b_jg_{ji}) \wedge a_{i+1} \wedge \ldots \wedge a_m \wedge b_dr_d.$$

Both sides of this equation contain m-exterior products of m-vectors; the products, therefore, are determinants and hence just field elements. Also, the products are skew-symmetric so that $a_c$ on the LHS and $b_d$ on the RHS of the equation may be moved into position between $a_{c-1}$ and $a_{c+1}$ while retaining the validity of the equation. Thus we can get the arbitrary closed-loop transfer function $y_c/r_d$ with a single loop closure from the output $y_i$ to the input $u_j$ through the feedback element $g_{ji}$, as

$$\frac{y_c}{r_d} = \frac{a_1 \wedge \ldots \wedge a_{c-1} \wedge b_d \wedge a_{c+1} \wedge \ldots \wedge a_{i-1} \wedge (a_i + b_jg_{ji}) \wedge a_{i+1} \wedge \ldots \wedge a_m}{a_1 \wedge \ldots \wedge a_{c-1} \wedge a_c \wedge a_{c+1} \wedge \ldots \wedge a_{i-1} \wedge (a_i + b_jg_{ji}) \wedge a_{i+1} \wedge \ldots \wedge a_m}$$

or

$$\frac{y_c}{r_d} = \frac{\begin{matrix}(a_1 \wedge \ldots \wedge a_{c-1} \wedge b_d \wedge a_{c+1} \wedge \ldots \wedge a_m \\ + a_1 \wedge \ldots \wedge a_{c-1} \wedge b_d \wedge a_{c+1} \wedge \ldots \wedge a_{i-1} \wedge b_j \wedge a_{i+1} \wedge \ldots \wedge a_mg_{ji})\end{matrix}}{(a_1 \wedge \ldots \wedge a_m + a_1 \wedge \ldots \wedge a_{i-1} \wedge b_j \wedge a_{i+1} \wedge \ldots \wedge a_mg_{ji})}.$$

In order to interpret the above expression for the closed-loop transfer function $y_c/r_d$ observe that, in terms of numerical calculations, each of the two terms in the numerator and in the denominator of the above equation is simply a determinant; thus a total of four determinantal calculations is involved here. Determinants of this sort recur in transfer function expressions with one or more loops closed; they are called numerators of various kinds [6]. The kind of the numerator is dependent on the nature

of the mix between the columns of A and the columns of B in the particular determinantal expression. For instance, the above equation contains two numerators of the first kind, that is, two determinants resulting from the m-exterior product of (m-1) columns of A with one column of B, namely

$$N_{u_d}^{y_c} = a_1 \wedge \cdots \wedge a_{c-1} \wedge b_d \wedge a_{c+1} \wedge \cdots \wedge a_m$$

and

$$N_{u_j}^{y_i} = a_1 \wedge \cdots \wedge a_{i-1} \wedge b_j \wedge a_{i+1} \wedge \cdots \wedge a_m.$$

Similarly, the equation has one numerator of the second kind, that is, the m-exterior product of (m-2) columns of A with two columns of B, namely

$$N_{u_d u_j}^{y_c y_i} = a_1 \wedge \cdots \wedge a_{c-1} \wedge b_d \wedge a_{c+1} \wedge \cdots \wedge a_{i-1} \wedge b_j \wedge a_{i+1} \wedge \cdots \wedge a_m.$$

Then, under the condition that $\det A = \Delta$, we can express the closed-loop transfer function $y_c/r_d$ compactly as

$$\frac{y_c}{r_d} = \frac{N_{u_d}^{y_c} + g_{ji} N_{u_d u_j}^{y_c y_i}}{\Delta + g_{ji} N_{u_j}^{y_i}}.$$

Thus, in the one-loop case, that is, with exactly <u>one</u> feedback loop (from $y_i$ to $u_j$ through $g_{ji}$ as shown in Figure 3), the expression for the arbitrary closed-loop transfer function involves numerators of the first and <u>second</u> kinds. It has been found [8] by using a similar exterior algebraic mechanism for manipulating the closed-loop transfer function expression that in the k-loop case, that is, with feedback loops from k different outputs to k different comparison points, numerators up to the (k+1)th kind are involved. Or, the highest kind of numerator in the arbitrary closed-loop transfer function expression is intimately connected with the number of loop closures.

<u>The Numerator of the kth Kind and the kth Exterior Map</u>

The arbitrary numerator of the kth kind is formed, as an extension of the definition of numerators of the first and second kind, by taking the m-exterior product of (m-k) columns of A with k columns of B, as

$$N_{u_{j_1} \cdots u_{j_k}}^{y_{i_1} \cdots y_{i_k}} = a_1 \wedge \cdots \wedge a_{i_1-1} \wedge b_{j_1} \wedge a_{i_1+1} \wedge \cdots \wedge a_{i_k-1} \wedge b_{j_k} \wedge a_{i_k+1} \wedge \cdots \wedge a_m.$$

We will show in the following that numerators of the kth kind may be understood in the context of the induced kth exterior map for the input R(s)-vector space U to the output R(s)-vector space Y. That, and the fact that numerators of different kinds are intimately related to the number of loop closures, would lead to the conclusion that there exists a strong association between the induced exterior map and loop closures.

Consider the equation relating the output y to the input u, expressed as

$$y = A^{-1} B u.$$

The vectors u and y belong to R(s)-vector spaces U and Y of dimen-

sion p and m respectively. Thus we have the morphism of vector spaces

$$A^{-1}B : U \to Y.$$

We have seen earlier that such a morphism of vector spaces induces a morphism of algebras

$$(A^{-1}B)\hat{} : \Lambda U \to \Lambda Y$$

as shown below

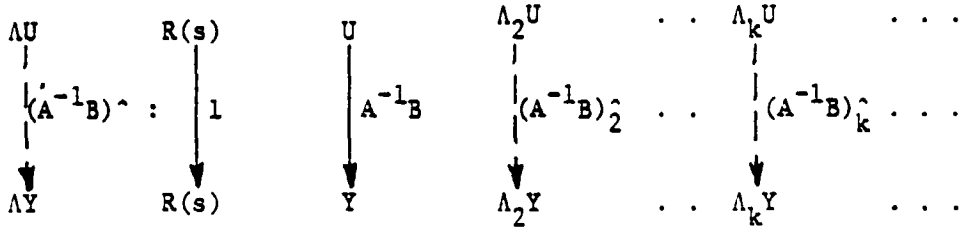



Figure 4. The Exterior Morphism Induced By $A^{-1}B$.

Of particular interest is the map

$$(A^{-1}B)\hat{}_k : \Lambda_k U \to \Lambda_k Y$$

because we intend to show how this map is related to numerators of the kth kind. In order to try and compute this map, we express [8]

$$(A^{-1}B)\hat{}_k = (A^{-1})\hat{}_k B\hat{}_k = (A\hat{}_k)^{-1}B\hat{}_k = \frac{\text{adj } A\hat{}_k}{\det A\hat{}_k} \cdot B\hat{}_k .$$

Let us assume for the moment that we can find a matrix T such that

$$T\,A\hat{}_k = I \det A.$$

The reason for this assumption will become clear in the following; but for the moment, let us rewrite the above expression as

$$\frac{T}{\det A} \cdot A\hat{}_k = I.$$

Comparing the above with the definition of adj $A\hat{}_k$ as

$$\frac{\text{adj } A\hat{}_k}{\det A\hat{}_k} \cdot A\hat{}_k = I ,$$

and because $A\hat{}_k$ is invertible, we have

$$\frac{\text{adj } A\hat{}_k}{\det A\hat{}_k} = \frac{T}{\det A} .$$

Thus we can replace adj $A\hat{}_k$/det $A\hat{}_k$ by T/det A in order to rewrite $(A^{-1}B)\hat{}_k$ as

$$(A^{-1}B)\hat{}_k = \frac{T\,B\hat{}_k}{\det A}.$$

The Nature of the Matrix T

We now investigate the nature of the matrix T; to understand T is to understand the nature of the kth exterior map $(A^{-1}B)\hat{}_k$, because $(A^{-1}B)\hat{}_k$ is just $TB_k$/det A. Recall that T has been defined in terms of

its action on the image of $A_k^{\wedge}$, as

$$T\, A_k^{\wedge} = I \det A.$$

Now $A_k^{\wedge}$ is calculated by forming all possible kxk minors from the matrix A, and is thus

$$A_k^{\wedge} = \begin{bmatrix} A_{1..k,1..k} & \cdots & A_{1..k,i_1..i_k} & \cdots & A_{1..k,\overline{m-k+1}..m} \\ \vdots & & \vdots & & \vdots \\ A_{\overline{m-k+1}..m,1..k} & \cdots & A_{\overline{m-k+1}..m,i_1..i_k} & \cdots & A_{\overline{m-k+1}..m,\overline{m-k+1}..m} \end{bmatrix}$$

where the overlines indicate grouping of terms, and the first and second set of subscripts of each element indicate the k rows and k columns, respectively, selected from the matrix A in order to form that particular kxk minor element. Expressing the matrix A as

$$A = [a_1 \cdots a_{i_1-1}\ a_{i_1}\ a_{i_1+1} \cdots a_{i_k-1}\ a_{i_k}\ a_{i_k+1} \cdots a_m],$$

the arbitrary ith column of $A_k^{\wedge}$ has been formed by selecting the k m-length columns $a_{i_1}, a_{i_2}, \ldots, a_{i_k}$ from A and forming all possible kxk minors from these k columns by choosing k rows at a time according to some predetermined convention. Thus the arbitrary ith column of $A_k^{\wedge}$ is of length $\binom{m}{k}$, the number of combinations of k items chosen at a time from m.

Based on Laplace's Theorem [9], a possible construction for T would be to make the ith row of T consist of cofactors corresponding to the kxk minor elements in the arbitrary ith column of $A_k^{\wedge}$. Thus the elements of the ith row of T are formed from the (m-k) complementary columns $a_1, \ldots, a_{i_1-1}, a_{i_1+1}, \ldots, a_{i_k-1}, a_{i_k+1}, \ldots, a_m$ of A. The (m-k) rows selected from these columns to form each element of the ith row T are complementary to the k rows selected from $a_{i_1}, \ldots, a_{i_k}$ to form the corresponding element in the ith column of A. Hence the order in which the (m-k) rows of A are selected in forming the elements of T may be said to follow a complementary convention to that used in selecting k rows when forming $A_k^{\wedge}$. Hence the matrix T may be represented as

$$T = \begin{bmatrix} \tilde{A}_{\overline{k+1}..m,\overline{k+1}..m} & \cdots & \tilde{A}_{1..\overline{m-k},\overline{k+1}..m} \\ \vdots & & \vdots \\ \tilde{A}_{\overline{k+1}..m,1..\overline{i_1-1}\ \overline{i_1+1}..\overline{i_k-1}\ \overline{i_k+1}..m} & \cdots & \tilde{A}_{1..\overline{m-k},1..\overline{i_1-1}\ \overline{i_1+1}..\overline{i_k-1}\ \overline{i_k+1}..m} \\ \vdots & & \vdots \\ \tilde{A}_{\overline{k+1}..m,1..\overline{m-k}} & \cdots & \tilde{A}_{1..\overline{m-k},1..\overline{m-k}} \end{bmatrix},$$

the "tilde" symbol denoting that the minors are appropriately signed.

Recall that we are interested in the product $T\, B_k^{\wedge}$. The matrix $B_k^{\wedge}$ is formed from the columns of the matrix B in the same manner as $A_k^{\wedge}$ was from the columns of A. B consists of p columns of length m, as

$$B = [b_1 \cdots b_{j_1-1}\ b_{j_1}\ b_{j_1+1} \cdots b_{j_k-1}\ b_{j_k}\ b_{j_k+1} \cdots b_p],$$

and $B_k^{\wedge}$ is given as

$$B_k^{\wedge} = \begin{bmatrix} B_{1..k,1..k} & .. & B_{1..k,j_1..j_k} & .. & B_{1..k,\overline{p-k+1..p}} \\ & & & & \\ B_{\overline{m-k+1..m},1..k} & .. & B_{\overline{m-k+1..m},j_1..j_k} & .. & B_{\overline{m-k+1..m},\overline{p-k+1..p}} \end{bmatrix}.$$

Here the arbitrary jth column of $B_k^{\wedge}$ is formed by picking the columns $b_{j_1},..,b_{j_k}$ from B and forming $\overline{kxk}$ minors using the same convention for the order in which k rows are chosen from m as before. Now consider a hybrid matrix

$$X_{ij} = [a_1 .. a_{i_1-1}\ b_{j_1}\ a_{i_1+1} .. a_{i_k-1}\ b_{j_k}\ a_{i_k+1} .. a_m]$$

and assume that we want to compute its determinant by invoking Laplace's Theorem. We may well select the k columns $b_{j_1},..b_{j_k}$ from $X_{ij}$ and from kxk minors to be multiplied by the corresponding (m-k) x (m-k) cofactors from the remaining columns $a_1,..,a_{i_1-1},a_{i_1+1},..,a_{i_k-1},a_{i_k+1},..,a_m$ of $X_{ij}$. But the former kxk minors are identical to the elements in the arbitrary jth column of $B_k^{\wedge}$, and the latter (m-k) x (m-k) cofactors are identical to the elements in the arbitrary ith row of T. The order of selection of k rows to form the former elements and of the (m-k) rows to form the latter elements are precisely complementary by convention, and thus the product of the ith row of T with the jth column of $B_k^{\wedge}$, that is, the ijth element of $T B_k^{\wedge}$, is just $\det X_{ij}$, that is,

$$a_1 \wedge \cdots \wedge a_{i_1-1} \wedge b_{j_1} \wedge a_{i_1+1} \wedge \cdots \wedge a_{i_k-1} \wedge b_{j_k} \wedge a_{i_k+1} \wedge \cdots \wedge a_m.$$

But the above is precisely the arbitrary numerator of the kth kind

$$N_{u_{j_1}..u_{j_k}}^{y_{i_1}..y_{i_k}}$$

so that we have

$$(A^{-1}B)_k^{\wedge} = \frac{T B_k^{\wedge}}{\det A} = \begin{bmatrix} N_{u_1..u_k}^{y_1..y_k} & .. & N_{u_{j_1}..u_{j_k}}^{y_1..y_k} & .. & N_{u_{p-k+1}..u_p}^{y_1..y_k} \\ \vdots & & \vdots & & \vdots \\ N_{u_1..u_k}^{y_{i_1}..y_{i_k}} & .. & N_{u_{j_1}..u_{j_k}}^{y_{i_1}..y_{i_k}} & .. & N_{u_{p-k+1}..u_p}^{y_{i_1}..y_{i_k}} \\ \vdots & & \vdots & & \vdots \\ N_{u_1..u_k}^{y_{m-k+1}..y_m} & .. & N_{u_{j_1}..y_{j_k}}^{y_{m-k+1}..y_m} & .. & N_{u_{p-k+1}..u_p}^{y_{m-k+1}..y_m} \end{bmatrix} / \det A.$$

Hence the kth exterior map, represented in matrix form, has the numerators of the kth kind as its elements.

## Conclusion

Whereas numerators of different kinds are used in closed-loop feedback design methods in industry via well-tested software packages, not

much is available in the literature about their nature other than that they are determinants involving columns of A and B; we have seen in the above section, however, that they are, in fact, the elements of the appropriate induced exterior map. In earlier sections we saw the connection between feedback loop closures and numerators of different kinds; thus, as the feedback loops are closed one by one, the different induced exterior maps come into play. Current multivariable feedback design methods attempt to shape multioutput responses based only on the last exterior map and, consistent with this, present definitions of multivariable zeros involving only minors of the largest order in the transfer matrix. However, because the intermediate exterior maps are also involved when we close loops, it would seem that we would be making better use of the available information in shaping multi-output responses if we defined intermediate zeros from all the maps when we try to set up a valid multivariable counterpart of the familiar SISO zeros.

## REFERENCES


1. V. Seshadri and M. Sain, "An Approach to Pole Assignment by Exterior Algebra," Proc. Fourteenth Allerton Conf. Circuit and System Thoery, pp. 399-407, 1976.

2. R.R. Gejji and M.K. Sain, "Application of Polynomial Techniques to Multivariable Control of Jet Engines," Proc. Fourth IFAC Symp. Multivariable Technological Systems, New Brunswick, Canada, 1977.

3. V. Seshadri and M.K. Sain, "An Application of Exterior Algebra to Multivariable Feedback Loops," Proc. 1979 Conf. Information Sciences and Systems, Baltimore, Maryland, pp. 337-342, 1979.

4. M.K. Sain, "The Growing Algebraic Presence in Systems Engineering: An Introduction," IEEE Proceedings, vol. 64, no. 1, pp. 96-111, 1976.

5. W.H. Greub, Multilinear Algebra, New York: Springer, 1967.

6. L.G. Hofmann, G.L. Teper and R.F. Whitbeck, "Application of Frequency Domain Multivariable Control Synthesis Techniques to an Illustrative Problem in Jet Engine Control," Intl. Forum Alternatives Linear Multivariable Control, Chicago, Illinois, 1977.

7. S. MacLane and G. Birkhoff, Algebra, London: Macmillan, 1967.

8. V. Seshadri, "Multivariable Feedback Loop Closures: An Approach by Exterior Algebra," Ph.D. dissertation, Department of Electrical Engineering, University of Notre Dame, Notre Dame, Indiana, October 1979.

9. F.E. Hohn, Elementary Matrix Algebra, New York: Macmillan, 1967.